US 20070288715A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2007/0288715 A1

Boswell et al. (43) Pub. Date: Dec. 13, 2007

(54) MEDIA PLAYER

(75) Inventors: **Jeremy Mayo Boswell**, Llantwit Major (GB); **Jonathan Mark Kendrick**, Shropshire (GB); **Timothy John Revell**, Wolverhampton West Midland (GB)

Correspondence Address:
CHRISTOPHER & WEISBERG, P.A.
200 EAST LAS OLAS BOULEVARD
SUITE 2040
FORT LAUDERDALE, FL 33301 (US)

(73) Assignee: **ROK PRODUCTIONS LIMITED**, Albrighton (GB)

(21) Appl. No.: **11/629,556**

(22) PCT Filed: **Jun. 14, 2005**

(86) PCT No.: **PCT/IB05/51965**

§ 371(c)(1),
(2), (4) Date: **Aug. 6, 2007**

(30) **Foreign Application Priority Data**

Jun. 14, 2004 (GB) 0413231.2
Oct. 26, 2004 (GB) 0423761.6
Feb. 25, 2005 (GB) 0504013.4

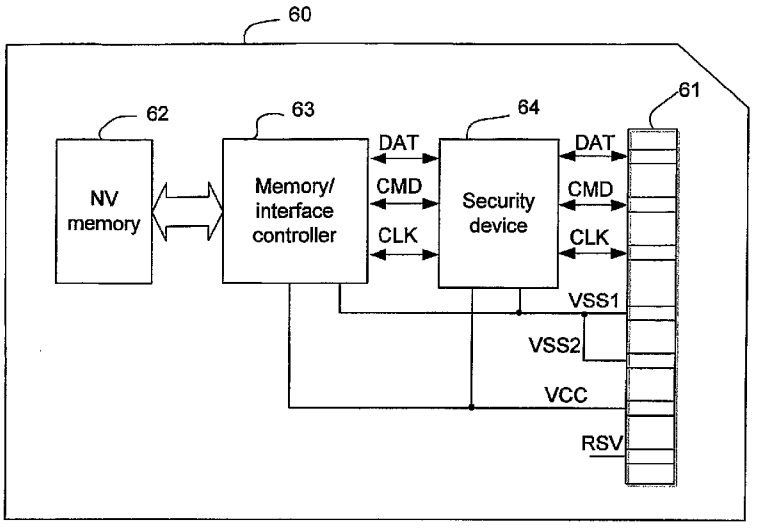
Publication Classification

(51) **Int. Cl.**
G06F 12/00 (2006.01)

(52) **U.S. Cl.** **711/164**

(57) **ABSTRACT**

Content is provided on a medium or is transmitted, for example streamed, with in built protection against unauthorised playback or copying. Video content data is provided in content blocks, each of which relates to a video frame and has a corresponding header. Some content clocks have a digital fingerprint interposed somewhere in the data comprising that content block. Thus, the amount of the data in the content block is increased by the addition of the digital fingerprint. Information identifying the length of the content block in the corresponding header is not altered. Some or all content blocks are obfuscated following the additional of the digital fingerprints. In a media player, de-obfuscation is performed, and digital fingerprints then removed before the resulting content blocks are decoded. Whilst listening to music streamed to a mobile terminal, a user can purchase the track currently being played. An automated extraction configuration module examines metadata stored on a DVD to determine the configuration of content data stored on it. Extraction configuration data from a memory area (**17**) is utilised by a DVD decryption and extraction module (**18**) to extract movie data from the DVD, which is written as AVI data to an intermediate format movie data area. A mobile format conversion module (**19**) converts movie data stored in the extracted movie data area (**14**) and provides a movie in mobile telephone consumable format in a mobile format movie data area (**20**). The data also includes two or more media players and a loader program. The mobile device is controlled to run the loader program initially. The program detects the relevant configuration of the mobile device and determines which of the media players to use to consume the movie content data. A portable data storage medium (**60**) includes non-volatile memory (**62**), and an interface (**61**) for connecting to an external device. It also includes a controller (**63**) which reads data out from the non-volatile memory. A security device (**64**) is connected between the controller and the memory.



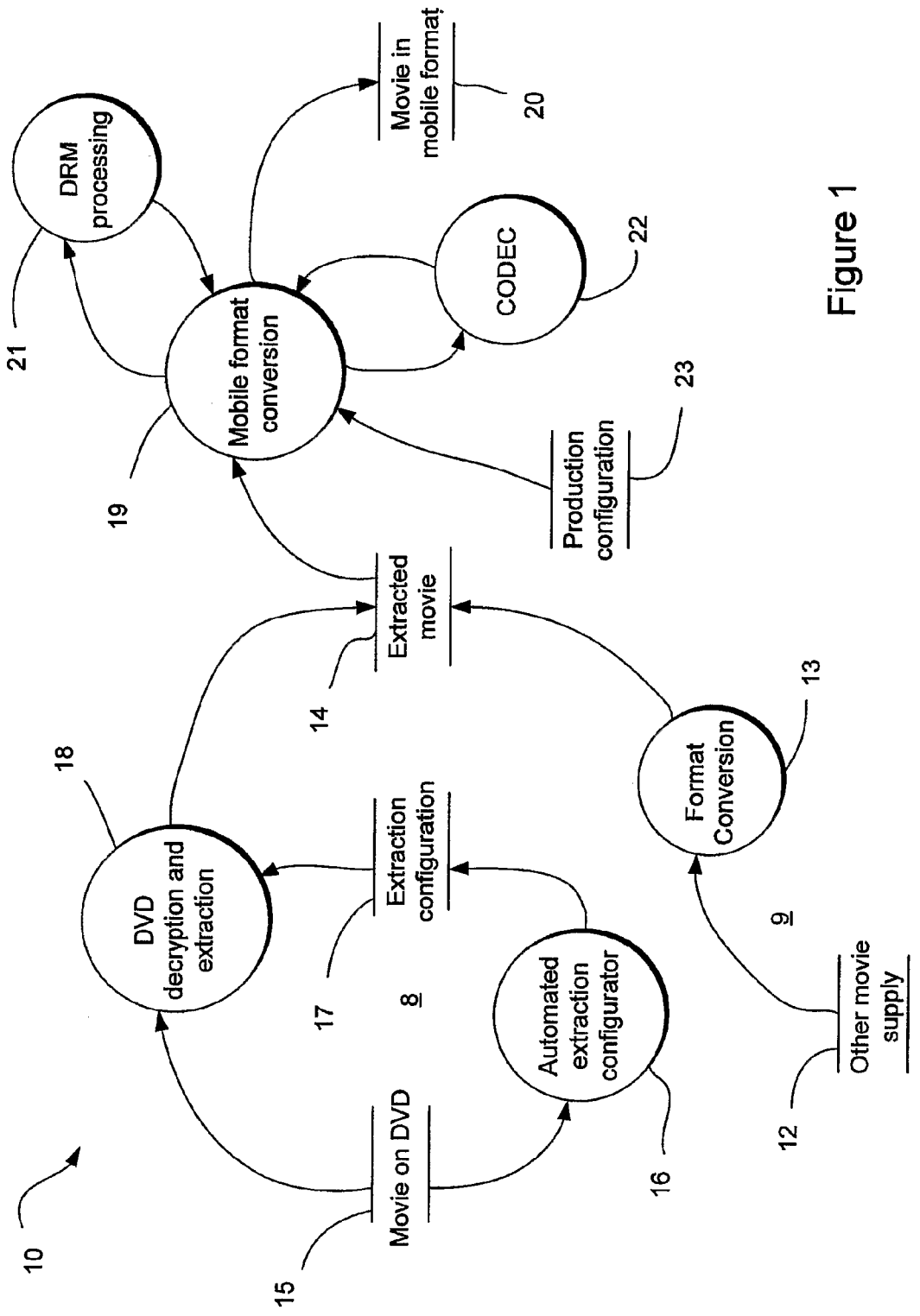

10
DVD decryption and extraction
18
Extraction configuration
17
8
Automated extraction configurator
16
Movie on DVD
15
Extracted movie
14
Mobile format conversion
19
DRM processing
21
Movie in mobile format
20
CODEC
22
Production configuration
23
Format Conversion
13
9
Other movie supply
12

Figure 1

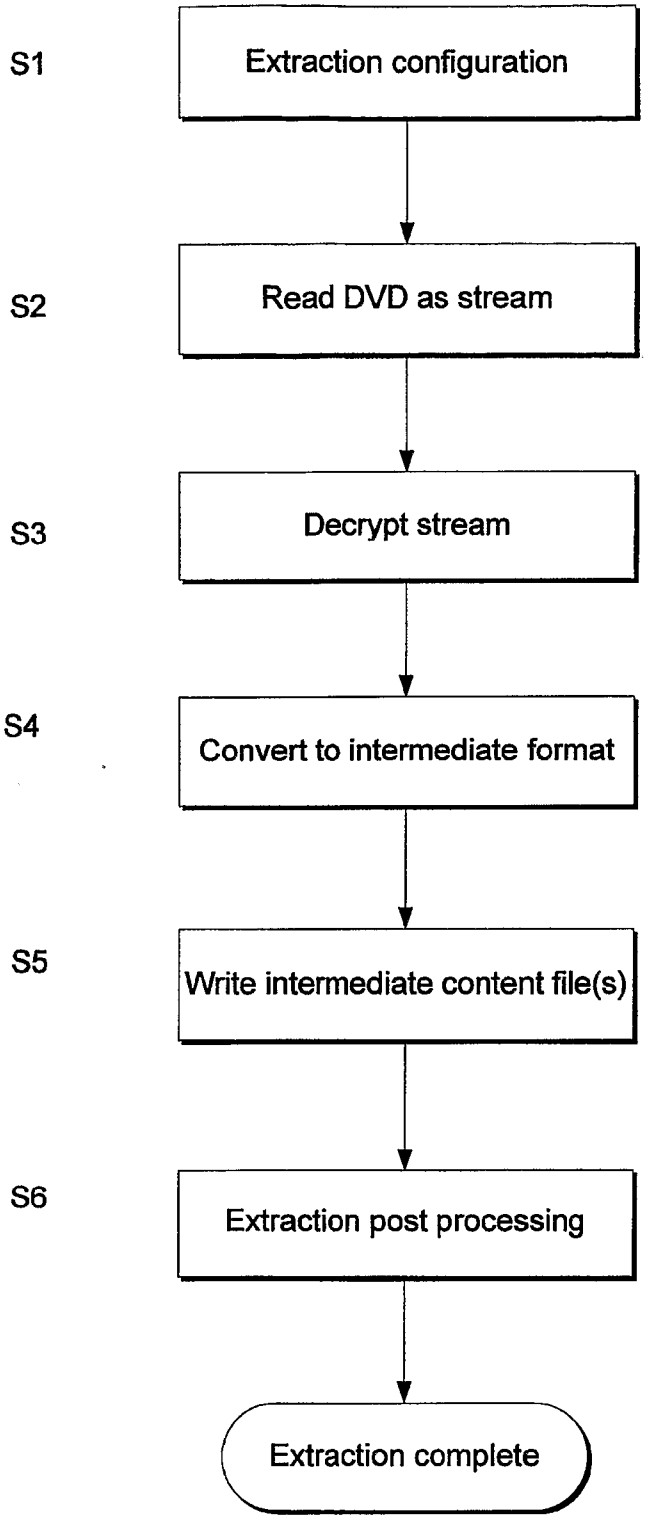

S1
Extraction configuration
S2
Read DVD as stream
S3
Decrypt stream
S4
Convert to intermediate format
S5
Write intermediate content file(s)
S6
Extraction post processing
Extraction complete

Figure 2

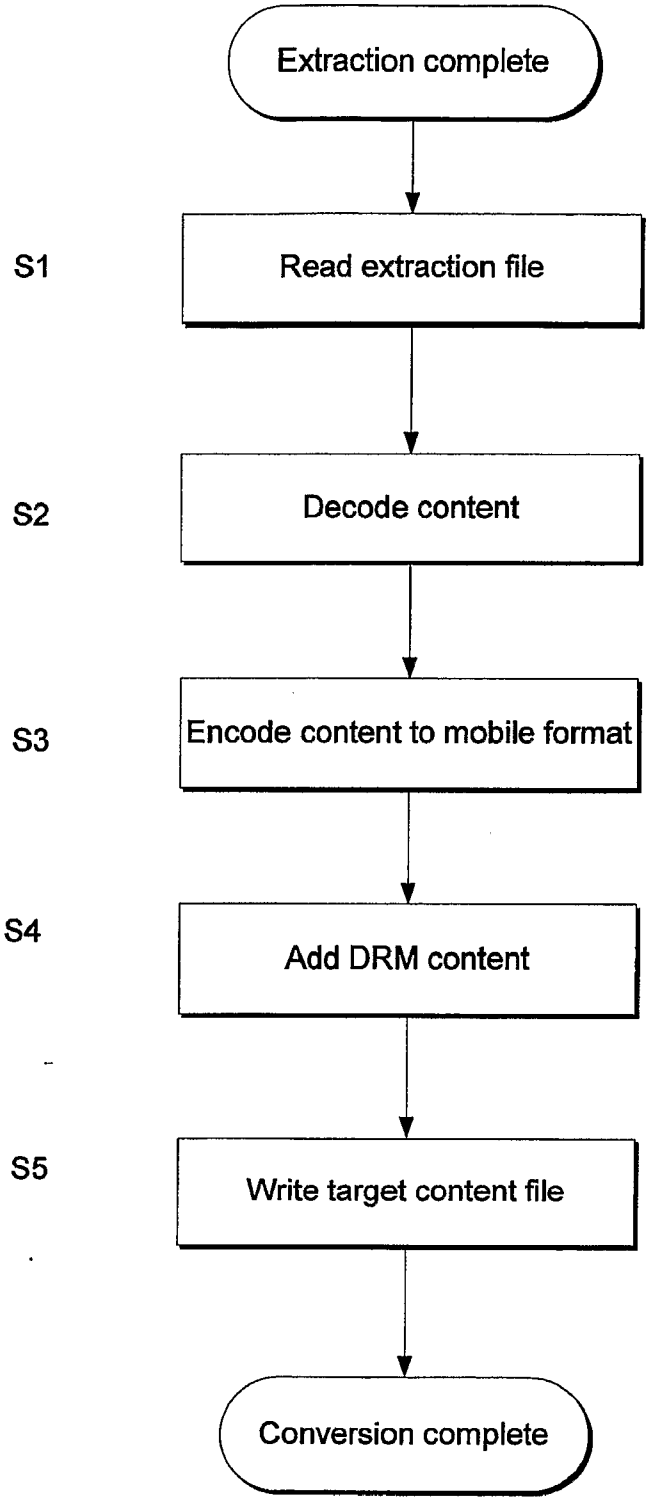

Extraction complete
S1
Read extraction file
S2
Decode content
S3
Encode content to mobile format
S4
Add DRM content
S5
Write target content file
Conversion complete

Figure 3

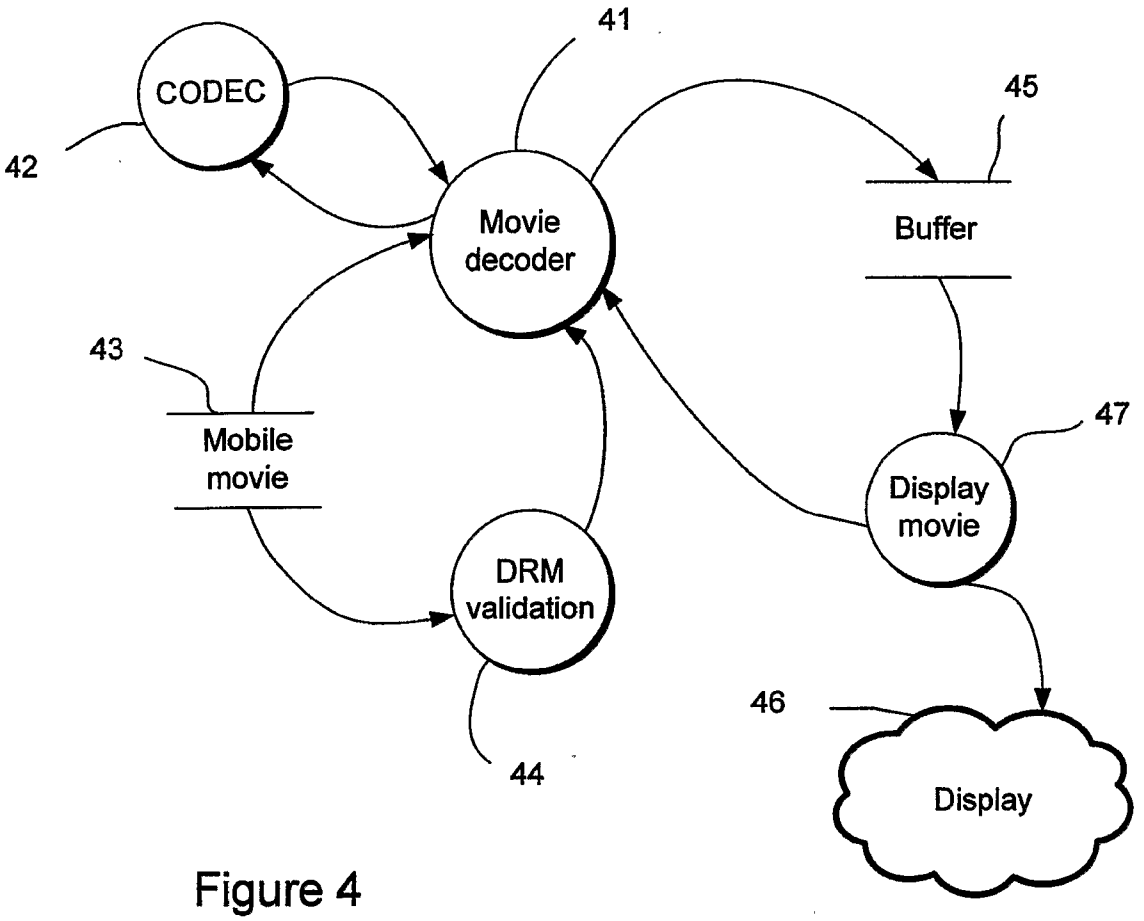

41
CODEC
42
45
Movie decoder
Buffer
43
Mobile movie
47
Display movie
DRM validation
46
44
Display

Figure 4

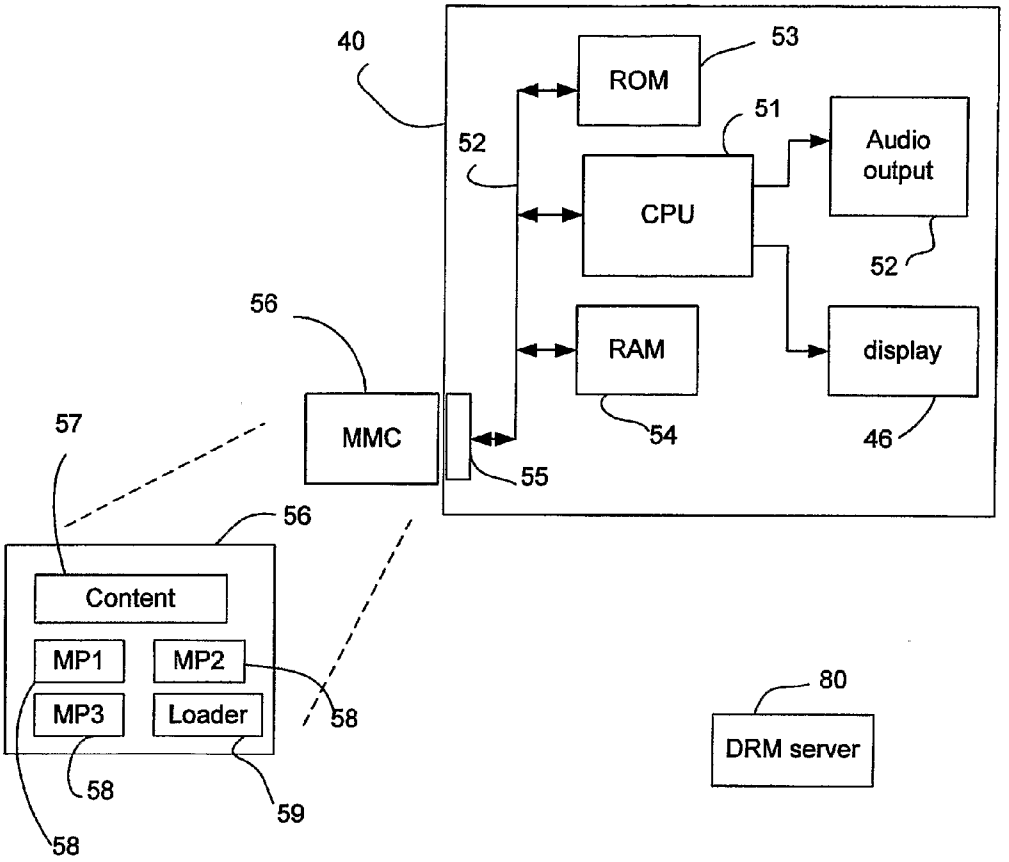

40
ROM
53
51
52
Audio output
CPU
52
56
RAM
display
MMC
54
46
57
55
56
Content
MP1
MP2
58
MP3
Loader
58
58
59
80
DRM server

Figure 5

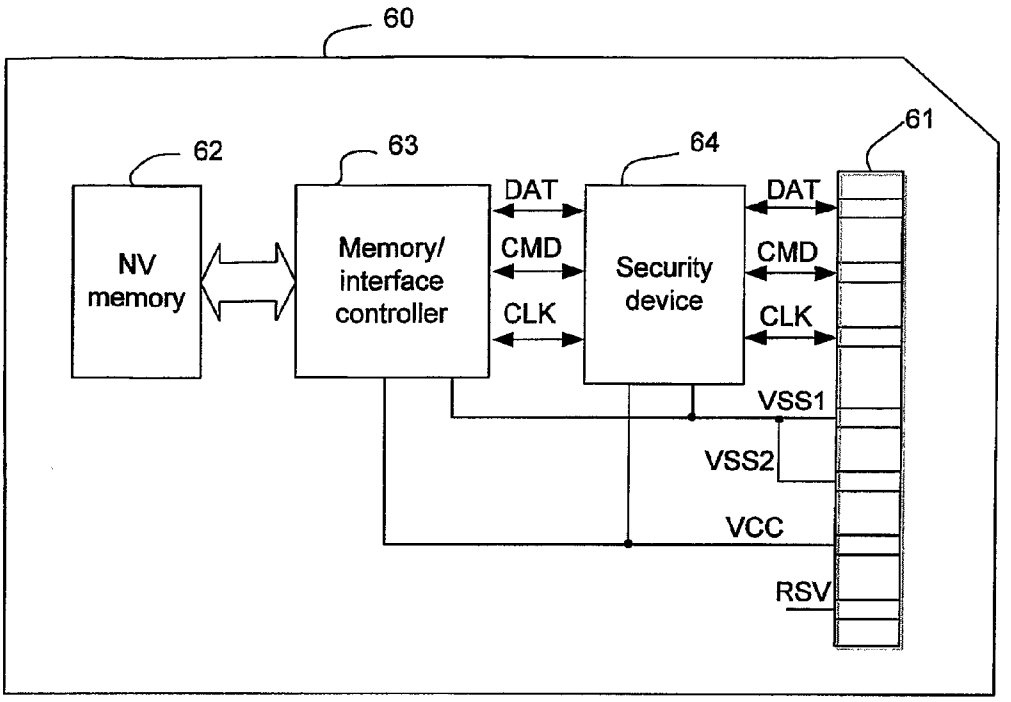


Figure 6

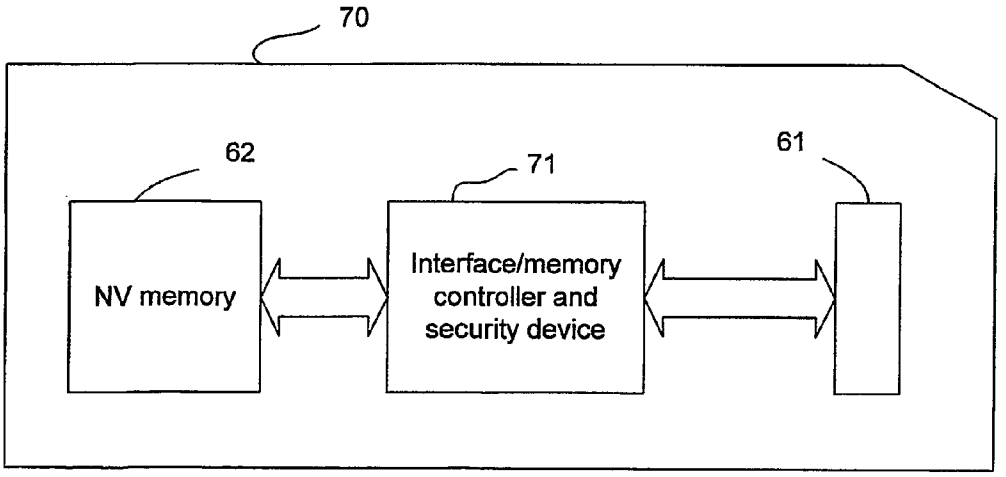


Figure 7

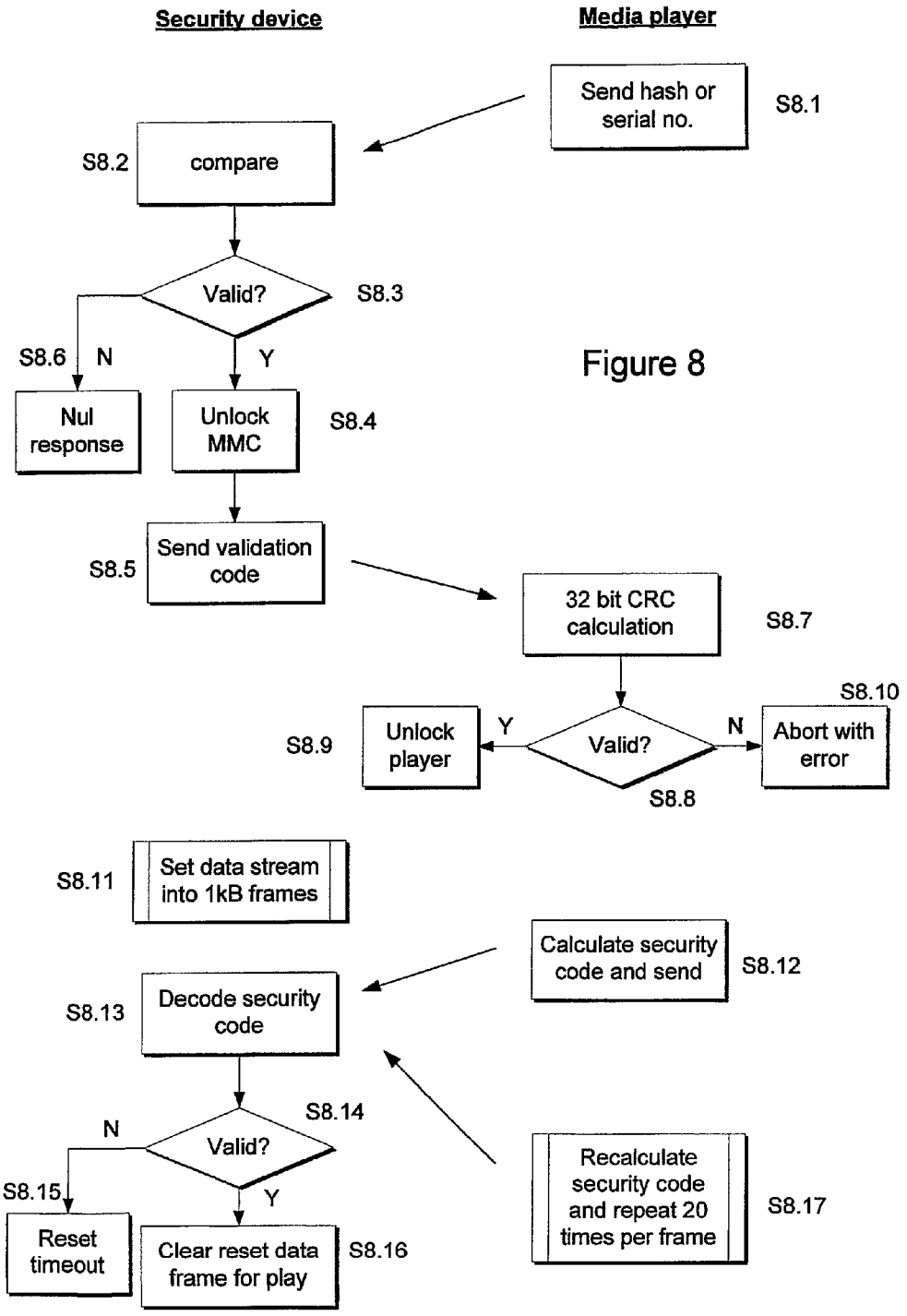

Security device
Media player
Send hash or serial no.
S8.1
S8.2
compare
Valid?
S8.3
S8.6
N
Y
Nul response
Unlock MMC
S8.4
S8.5
Send validation code
32 bit CRC calculation
S8.7
S8.10
S8.9
Unlock player
Y
Valid?
N
Abort with error
S8.8
S8.11
Set data stream into 1kB frames
Calculate security code and send
S8.12
S8.13
Decode security code
S8.14
N
Valid?
S8.15
Y
Reset timeout
Clear reset data frame for play
S8.16
Recalculate security code and repeat 20 times per frame
S8.17

Figure 8

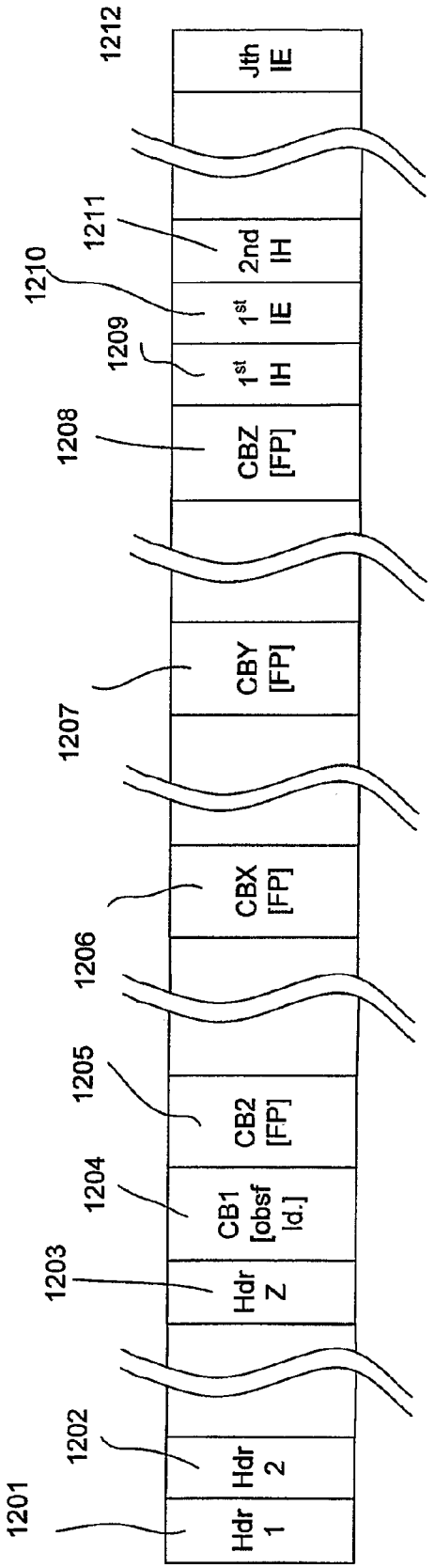

1201
Hdr 1
1202
Hdr 2
1203
Hdr Z
1204
CB1 [obsf Id.]
1205
CB2 [FP]
1206
CBX [FP]
1207
CBY [FP]
1208
CBZ [FP]
1209
1st IH
1210
1st IE
1211
2nd IH
1212
Jth IE
1200

Figure 9

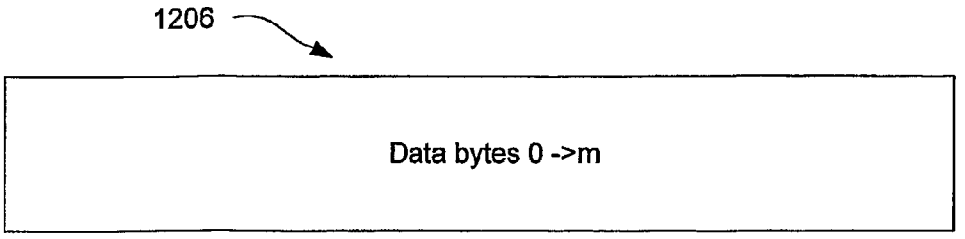


Figure 10A

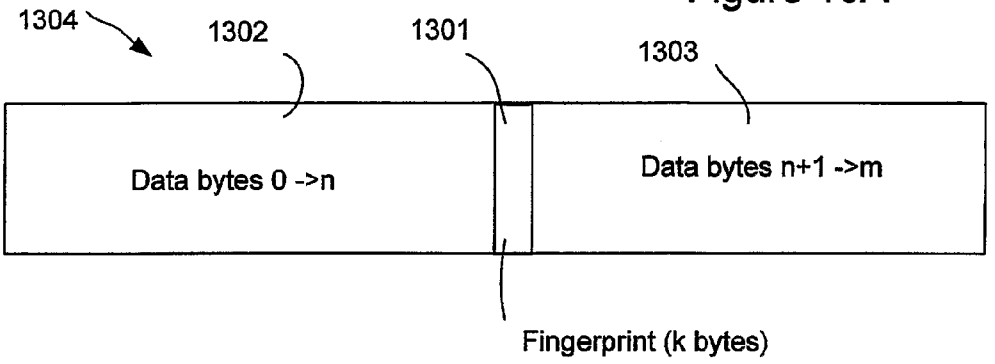


Figure 10B

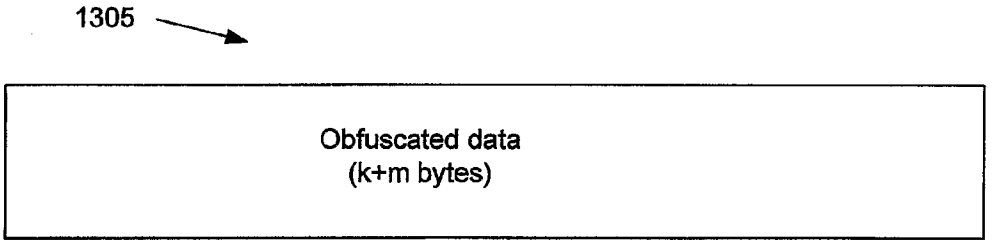


Figure 10C

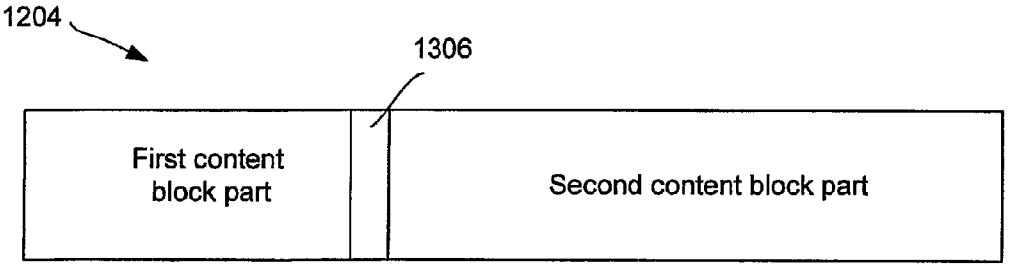


Figure 10D

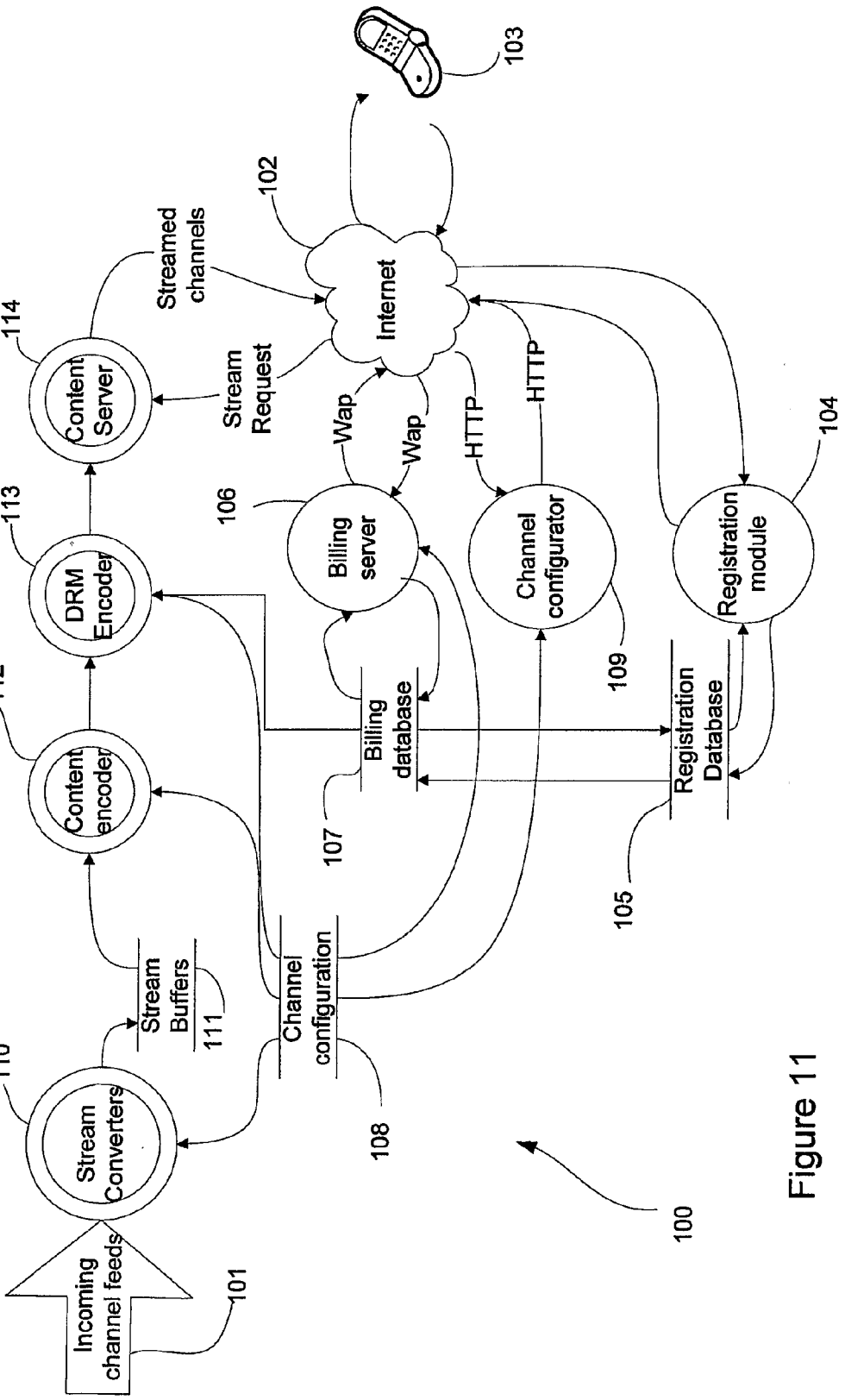

Incoming channel feeds
101
Stream Converters
110
Stream Buffers
111
Content encoder
112
DRM Encoder
113
Content Server
114
Streamed channels
Stream Request
Internet
102
103
Wap
Wap
HTTP
HTTP
Billing server
106
Channel configurator
109
Registration module
104
Billing database
107
Registration Database
105
Channel configuration
108
100

Figure 11

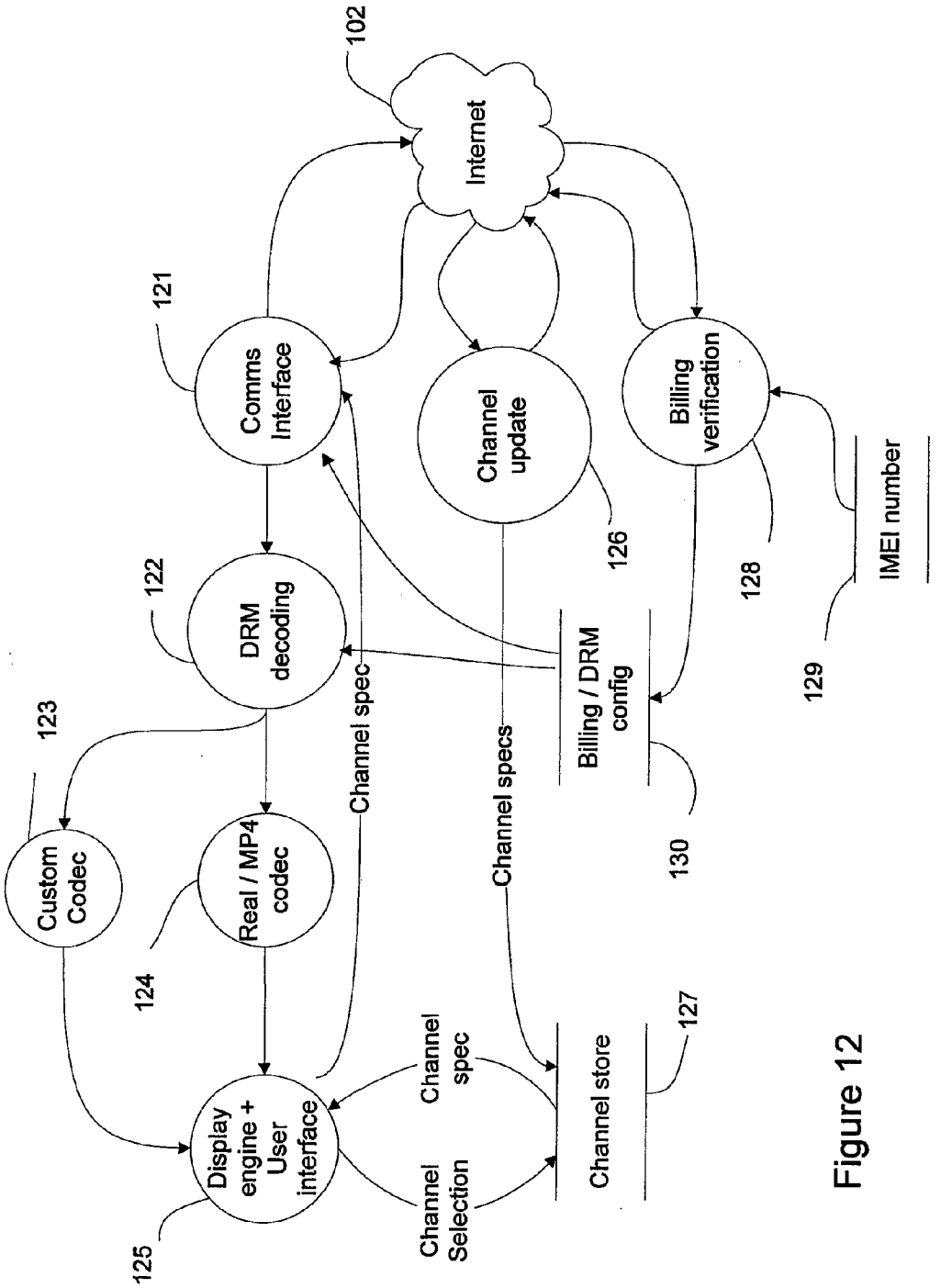

102
Internet
121
Comms Interface
Channel update
Billing verification
122
DRM decoding
126
128
IMEI number
129
123
Channel spec
Billing / DRM config
Channel specs
130
Custom Codec
Real / MP4 codec
124
Channel spec
Channel store
127
Display engine + User interface
Channel Selection
125

Figure 12

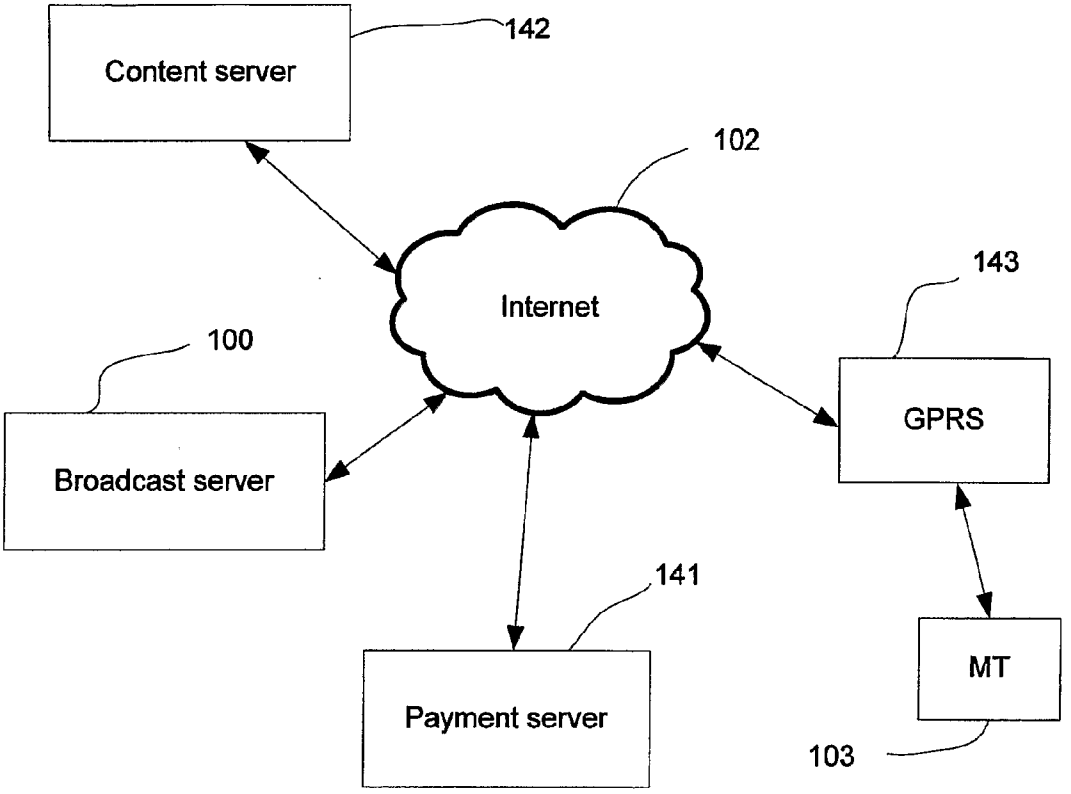

Content server
142
102
Internet
143
100
GPRS
Broadcast server
141
MT
Payment server
103

Figure 13

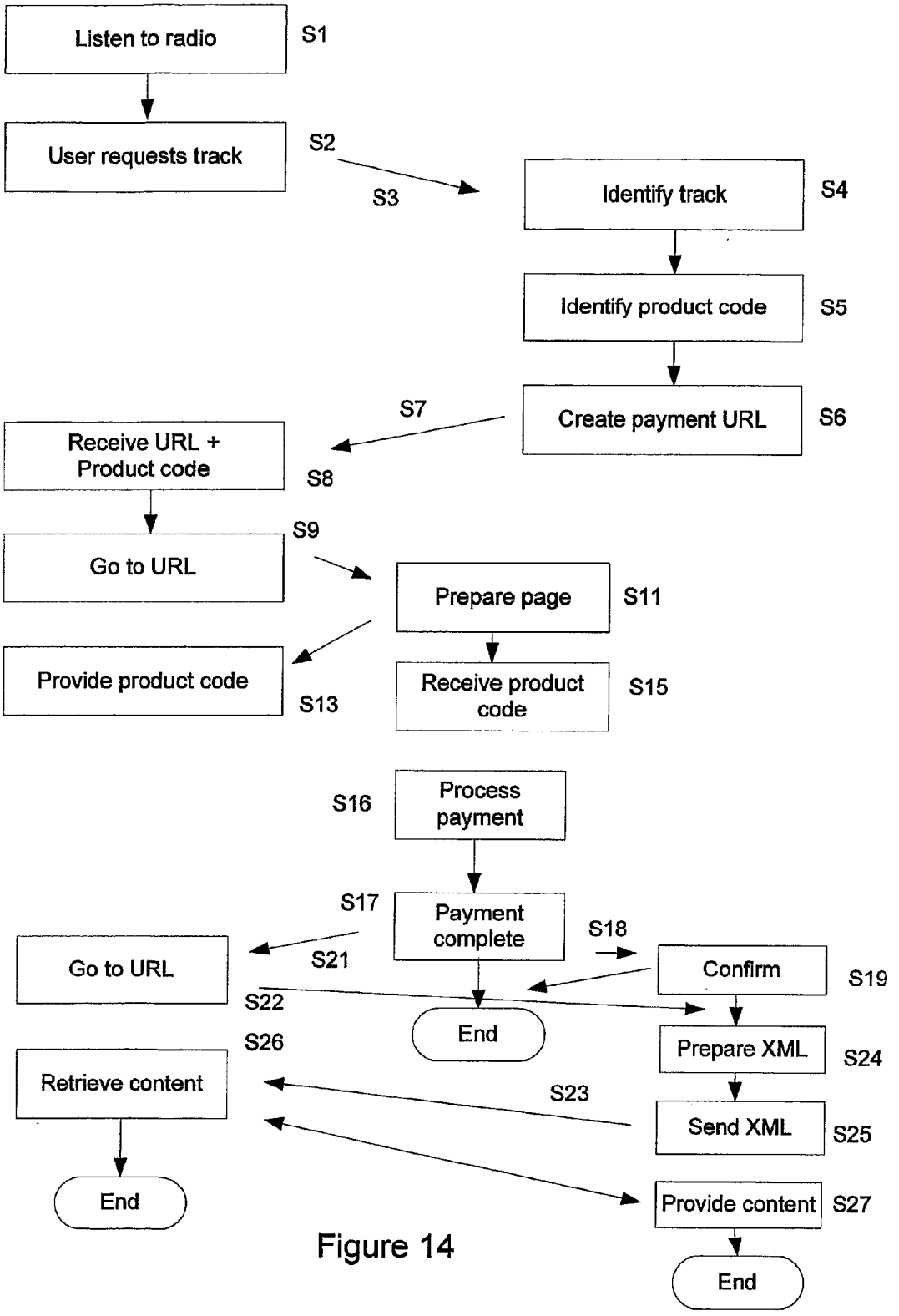

Listen to radio
S1
User requests track
S2
S3
Identify track
S4
Identify product code
S5
Create payment URL
S6
S7
Receive URL + Product code
S8
Go to URL
S9
Prepare page
S11
Provide product code
S13
Receive product code
S15
S16
Process payment
S17
Payment complete
S18
Go to URL
S21
Confirm
S19
S22
End
S26
Prepare XML
S24
Retrieve content
S23
Send XML
S25
End
Provide content
S27
End

Figure 14

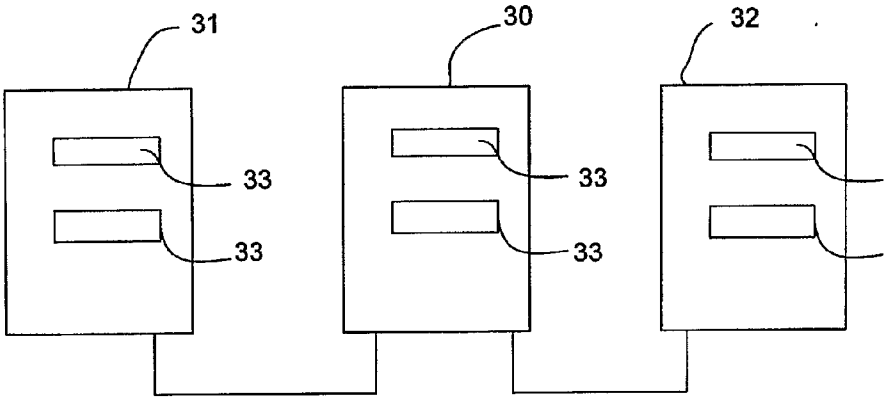


Figure 15

MEDIA PLAYER

[0001] Providing content subject to copyright on portable storage media and the like for playback on mobile devices introduces the possibility of interested persons being able to make illegal copies of the content, for distribution without the consent of, or payment to, the owner of the copyright.

[0002] It is known to stream coded video and audio data to mobile devices, for decoding and playback thereon. One such service is operated by Sprint PCS Vision Multimedia Services. However, supplying content which is subject to copyright in this way can make it vulnerable to being intercepted by interested third parties, even when GPRS or a 3G data service is used for delivery. Content so provided could also be recorded, by persons with suitable technical proficiency, in a form in which it could be used for unauthorised copying of the content.

[0003] It is conventional to use encryption to protect content which is subject to copyright. In this way, only recipients provided with a correct encryption key can decrypt the content and consequently decode it properly for playback. Weak protection can be obtained using relatively short keys. Stronger protection requires longer encryption keys to be used. However, decryption places a processing overhead on the receiver. In the case of mobile devices, the process of decryption results in the consumption of significant amounts of electrical power, thereby decreasing the time it takes for the battery to discharge to a level which is insufficient to continue powering the mobile device. Stronger encryption requires more processing to decrypt the same amount of content data, thereby exasperating the problem. Encryption of content does not prevent recording, by persons with suitable technical proficiency and the decryption key or keys, of decrypted content in a form in which it could be used for unauthorised copying of the content.

[0004] Thus, the inventors have perceived a need for the protection of content which can place less processing overhead on devices which are intended to decrypt and playback the content. The inventors have also perceived a need for the protection of content from persons who may be interested in authorised copying of the content.

[0005] According to the invention, there is provided a media player operable to decode content data, the media player being arranged to use excess data location information to identify the location of each of plural excess data items within content blocks forming part of the content data, to remove the excess data from the locations so determined, and to decode the content data following removal of the excess data.

[0006] This can allow playback of the content to be limited to media players which are authorised to playback the content. Furthermore, authorisation can be made content item specific or content generic. Other advantages will be appreciated from the description of the embodiments described below.

[0007] Preferably, the excess data location information is obtained over a channel separate from a source of the content data. This allows control over which media players are allowed to playback the content, since the excess data location information is separated from the content data. Furthermore, the ability to playback the content can be limited to media players held by users whom have paid for or otherwise obtained a license to playback the content.

[0008] Further preferably, each occurrence of the excess data is a digital fingerprint. In this way, the excess data can be such that it is very difficult or impossible to identify within the content data, yet is able to be verified as the excess data by a media player that has the excess data location information.

[0009] Advantageously, the media player is arranged to determine whether data forming part of an authorisation code includes data corresponding to the digital fingerprint, and to decode the content data only in response to a positive determination. This can allow further protection of the content data since the media player will play back the content only if the location of the fingerprints and the fingerprint is known. This leads to the further advantage that a single excess data location pattern can be used with different digital fingerprints. In particular, the content can be allowed to be played back by one or more media players which have the excess data location information and the fingerprint, whilst disallowing playback by one or more media players which have the excess data location information but do not know the fingerprint. The authorisation code may be a media code. A media code may additionally include information allowing playback to be restricted to a particular mobile device, for instance by its IMEI number.

[0010] If each of the digital fingerprints comprises a data sequence which is the same for each occurrence of the digital fingerprint, the locations of the fingerprints can be verified by a media player on the basis of an unlock code or other means including less data than for a corresponding system in which numerous different fingerprints were present in the content.

[0011] Preferably, the media player is arranged to determine whether data forming part of an authorisation code includes data corresponding to a serial number of the media player, and to decode the content data only in response to a positive determination. This allows the use of authorisation codes which are specific to a particular media player. Thus, an authorisation code obtained by user who has purchased or otherwise obtained a license for playback of content on their device cannot be used also by other media players, thereby limiting content playback to a single media player.

[0012] Advantageously, the media player is operable to derive the excess data location information from the authorisation code. This allows the media player to obtain all or most of the information it needs to playback the content from a single authorisations code, which provides a substantial amount of protection for the content whilst allowing authorisation of a media player to playback the content through a relatively simple authorisation process.

[0013] The media player can be operable to use preprogrammed information to identify content blocks in which the excess data is located, and to use the excess data location information to determine the location of the excess data within those content blocks. This allows effective content protection whilst allowing a relatively simple function in the media player to playback the content correctly when authorised to do so. The amount of data needed to achieve this can be quite small, allowing authorisation to be carried out over limited channels, for instance SMS.

[0014] Embodiments of the present invention will now be described by way of example with reference to the accompanying drawings, in which:

[0015] FIG. **1** is a schematic diagram of audio-visual content provision apparatus useful in understanding the invention;

[0016] FIGS. **2** and **3** are flowcharts illustrating steps of operation of the FIG. **1** apparatus;

[0017] FIG. **4** is a schematic drawing illustrating apparatus for playback of the converted audio-visual content in a mobile telephone according to the invention;

[0018] FIG. **5** illustrates a combination of an MMC and the mobile telephone of FIG. **4**;

[0019] FIGS. **6** and **7** illustrate alternative embodiments of MMC hardware, useful in understanding the invention;

[0020] FIG. **8** is a flowchart illustrating security validation between the mobile telephone of FIG. **4** and the MMC of FIG. **6** or FIG. **7**;

[0021] FIG. **9** shows content stored on an MMC, and is useful in understanding the invention;

[0022] FIGS. **10**A, **10**B, **10**C and **10**D show content blocks stored in the FIG. **9** MMC, their creation and their use in a media player according to the invention;

[0023] FIG. **11** is a schematic diagram illustrating components of a broadcast server operable with a media player according to the present invention; and

[0024] FIG. **12** is a schematic block diagram illustrating components of a media player operable according to the present invention; and

[0025] FIG. **13** is a schematic diagram of components of a system through which a terminal is able to obtain content and operate according to the invention.

[0026] Throughout the drawings, reference numerals are re-used for like elements.

[0027] Referring firstly to FIG. **1**, content extracting and converting apparatus **10** is illustrated schematically. Two alternative sources of audio-visual content **8**, **9** are included. A first content source **8** utilises film or movie data stored on a DVD (digital video disk or digital versatile disk) **15**. An automated extraction configuration module **16** examines metadata stored on the DVD **15** to determine the configuration of content data stored on the DVD. This involves the application of a tcprobe, and an analysis of the information returned from the DVD **15**. This is described in more detail below. The result is data stored in an extraction configuration memory area **17** representing an extraction configuration. The extraction configuration data from the memory area **17** is utilised by a DVD decryption and extraction module **18** to extract movie data (i.e. the content data) from the DVD **15**. The result is content data in an intermediate format, which is written to an intermediate format movie data area **14**. The data included in the intermediate format movie data area **14** is in predetermined format and is suitable for conversion into a form ready for reproduction on a mobile telephone (not shown). Preferably the intermediate format is AVI. This format has the advantage of high resolution, yet is relatively easy to handle and it is relatively easy to convert from AVI into 3GPP and many other formats suitable for use by mobile devices.

[0028] The second source of audio-visual content **9** receives from a movie data storage area **12** data representing a movie (or film) in AVI (audio-visual interleave) or other format. The movie so supplied is converted by a format conversion module **13** before being written to the intermediate format movie data area **14**.

[0029] Thus, either of the audio-visual content sources **8**, **9** can be used to provide movie data in the intermediate format movie data area **14**.

[0030] A mobile format conversion module **19** converts movie data stored in the extracted movie data area **14** and provides a movie in mobile telephone consumable format in a mobile format movie data area **20**. The mobile format conversion module **19** utilises a digital rights management (DRM) processing module **21**, which allows certain control over the access and distribution of the resulting movie data. The conversion effected by the mobile format conversion module **19** uses a codec **22**, which preferably is custom-designed for the purpose. Importantly, the conversion effected by the mobile format conversion module **19** uses information stored in a production configuration data area **23**. By controlling the mobile format conversion module **19** on the basis of information specific to the configuration of, and thus tailored to, a target device, the apparatus **10** can be used to provide movie data for any of potentially a large number of target mobile devices.

[0031] The extraction effected by the audio-visual content source **12** will now be described in detail with reference to FIG. **2**.

[0032] In FIG. **2**, extraction configuration is effected at step S**1**. This utilises the automated extraction configuration **16** shown in FIG. **1**. Extraction configuration commences by analysing the DVD source **15**. The result of an example analysis, i.e. what is returned in response to a query, is illustrated below:

(dvd_reader.c) mpeg2 pal 16:9 only letterboxed U0 720×576 video

(dvd_reader.c) ac3 en drc 48 kHz 6Ch

(dvd_reader.c) ac3 de drc 48 kHz 6Ch

(dvd_reader.c) ac3 en drc 48 kHz 2Ch

(dvd_reader.c) subtitle 00=<en>

(dvd_reader.c) subtitle 01=<de>

(dvd_reader.c) subtitle 02=<sv>

(dvd_reader.c) subtitle 03=<no>

(dvd_reader.c) subtitle 04=<da>

(dvd_reader.c) subtitle 05=<fi>

(dvd_reader.c) subtitle 06=<is>

(dvd_reader.c) subtitle 07=<en>

(dvd_reader.c) subtitle 08=<de>

[tcprobe] summary for /media/dvdrecorder/, (*)=not default, 0=not detected

import frame size: -g 720x576 [720x576]

[0033] aspect ratio: 16:9 (*)

[0034] frame rate: -f 25.000 [25.000] frc=3

[0035] audio track: -a 0 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[0036] audio track: -a 1 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[0037] audio track: -a 2 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[tcprobe] V: 185950 frames, 7438 sec @ 25.000 fps

[tcprobe] A: 116.22 MB @ 128 kbps

[tcprobe] CD: 650 MB|V: 533.8 MB @ 602.0 kbps

[tcprobe] CD: 700 MB|V: 583.8 MB @ 658.4 kbps

[tcprobe] CD: 1300 MB|V: 1183.8 MB @ 1335.1 kbps

[tcprobe] CD: 1400 MB|V: 1283.8 MB @ 1447.9 kbps

[0038] This information is returned by tcprobe, which is part of transcode. Part of the extraction configuration process of S1 involves determining the configuration of the target device, which is represented by the information stored in the production configuration data area **23**. It is helpful therefore to understand the information that is stored there.

[0039] Information data stored in the production configuration data area **23** identifies the aspect ratio of the display of the target device. In most cases, the aspect ratio 4:3, although this may vary form device to device. Certain devices will include 16:9 (widescreen) aspect ratios, although in practice the aspect ratio may take a value which is not the same as a conventional television aspect ratio. The production configuration data also identifies the audio language required. It also identifies whether or not subtitles are required. If they are required, the production information-configuration identifies the language that the subtitles are required to be in. The bitrates of the video and the audio tracks are included in the production configuration data. The bitrates may depend on the capabilities of the target device, on the particular media player installed in the target device or on any other factors. The production configuration data may also indicate a maximum volume size, for example indicating the amount of usable memory in an MMC. The production configuration information also includes an indication of the format on which the movie data is to be stored. For example, this format can be 3GPP or MPEG-4 format, or any other suitable format.

[0040] The information included in the production configuration data area **23** also includes the type of the target device. This may be, for example, a model number of the mobile telephone on which the movie is to be reproduced. In some circumstances, it may be possible that two different mobile telephones having the same model number can have different hardware and/or software configurations. Where different configurations are possible, and this may have a bearing on the optimum processing effected by the apparatus **10**, the information stored in the production configuration data area **23** preferably also includes details of how the hardware and/or software configuration departs from the standard configuration, or perhaps instead merely specifies the configuration.

[0041] The automated extraction configuration module **16** determines from the information returned by tcprobe, (in particular the first line thereof reproduced above) that the DVD **15** contains only widescreen (that is 16:9 aspect ratio) video in MPEG 2 PAL format. The module **16** also determines that there are three audio tracks, identified by the second and fourth lines respectively. The first and second tracks have 6 channels each and 48 kHz sampling rates. The first is in the English language and the second is in the German language, as identified by the "en" and "de" designations. The third audio track is in the English language and is a stereo (two channel) signal having a 48 kHz sampling rate. The module determines also that the DVD **15** has eight subtitle tracks, in various languages. The module **16** also determines the frame rate, the number of frames and the length of the movie. The module **16** uses the last four lines of the returned information to determine the content bitrate variations that can be extracted from the DVD **16**.

[0042] The function of the automated extraction configuration module **16** also includes obtaining decryption keys, which are needed to allow the audio-visual content on the DVD to be reproduced.

[0043] The information determined by the automated extraction configuration module **16** constitutes the configuration of the DVD **15**.

[0044] Based on the information in the production configuration data area **23** and on the DVD configuration information, the automated extraction configuration module **16** makes a decision as to which audio tracks, which video channel (if there is more than one video channel) and which subtitle track are needed. Typically, the subtitle track identified by this process is the first listed subtitle track which is in the same language as the subtitle language identified in the production configuration data area **23**. Also, the audio track identified by this process is the audio track which is in the same language as the audio language identified in the production configuration data area **23** and which is most suitable for use by the target device. In most cases, Dolby™ Pro Logic™ audio channels will not be suitable, because most target devices will not be equipped to handle such audio signals. A stereo audio track will in most cases be the most suitable audio track, although any mono track may be most suitable for a target device with only mono audio capabilities. The video channel selected by this process typically is the main channel, i.e. the actual movie, and not any 'additional features', such as trailers and behind-the-scene documentaries and the like that are commonly included on DVDs. Data identifying the tracks and channels identified by this process is stored in the extraction configuration data area **17**.

[0045] In step S2, the data stored on the DVD **15** is read as a stream. This is represented by the arrow between the movie on DVD data area **15** and the DVD decryption and extraction module **18** in FIG. **1**. It is only the content which is read at this time, since the configuration information, or metadata, is not used by the DVD decryption and extraction module **18** directly. Also, it is only the relevant content which is read. The relevant content is identified to the DVD decryption and extraction module **18** by the information

stored in the extraction configuration data area **17**, which identifies the relevant video channel, the relevant audio channel and any relevant subtitle channel. At step S**3**, the relevant portions of the DVD data stream are decrypted by the DVD decryption and extraction module **18**. This decryption uses transcode with the keys extracted by the automated extraction configuration module **16**. Decryption is performed "on the fly", i.e. as a continuous process as the content is read from the DVD **15**. As the data is decrypted, it is converted into the intermediate format, i.e. AVI format. At step S**5**, the movie data is written into the extracted movie data buffer **14** as a file or series of files in the intermediate format.

[**0046**] At step S**6**, extraction post-processing is performed. This involves splitting or joining the content file or files present in the extracted movie data buffer **14** into components. Whether there is any splitting or any joining and the extent of it depends on the target device configuration information stored in the production configuration data area **23**. In most cases, this step will involve splitting the extracted content cleanly to multiple volumes. Providing movie content in the form of multiple volumes is desirable in many circumstances due to the limitations of mobile telephones. It is a fairly straightforward procedure to split DVD movie content into volumes corresponding to the DVD chapters present on the original DVD **15**. Following step S**6**, the extraction of the movie data is complete.

[**0047**] The result is movie data stored in the extracted movie data buffer **14** which is encoded into an intermediate format (e.g. AVI format) and which includes only one audio track, which is in the required language identified by the production configuration information stored in production configuration data area **23**, and optionally one subtitled track, in the required language. The extracted movie data typically is divided into a number of volumes, although this may not be necessary depending on the configuration of the target device.

[**0048**] Instead of using a DVD data source **15**, the other movie data storage area **12** may be used. In this. case, format conversion to the intermediate format, for example AVI, is carried out by the format conversion module **13**. If only DVD sources **15** will be used, then the second content source **9** can be omitted. If included, the format conversion module **13** takes a form which is suitable for the particular type of content provided at the other movie data storage area **12**. A separate format conversion module **13** may be needed for each type of data that can be stored in the other movie data storage area **12**.

[**0049**] The procedure of FIG. **3** begins with the extraction process complete. At step S**1**, the extraction file is read. This is an "on the fly" procedure and is represented by the arrow linking the extracted movie data buffer **14** with the mobile format conversion module **19**. At step S**2**, the mobile format conversion module **19** decodes the content comprising the movie data. The step uses transcode. At step S**3**, the decoded content is encoded into the required mobile format, as identified by the production configuration information stored in the production configuration data area **23**. The encoding is performed by the codec **22**. The encoding is performed in such a way as to result in audio and video content having the most appropriate bitrates. What are the most appropriate bitrates is determined by the mobile format conversion module **19**. In particular, the mobile format conversion module **19** uses knowledge of the number of video frames in the video data and the length of the audio track along with the maximum volume size information stored in the production configuration data area **23** to determine the most suitable bitrates. In most cases, the most suitable bitrates for the audio and video will be the bitrates which are the maximum possible bitrates which could be used to fit the entire content within the maximum volume size.

[**0050**] Usually, the bitrates selected for the audio and the video give rise to comparable quality for those components, although there can be some discrepancy if this results in mobile format movie data which would give an improved playback experience if this is possible having regard to the maximum volume size. For example, if audio and video content at a certain quality level would give rise to data exceeding the maximum volume size but that content at a quality level immediately below that would give rise to a significant shortfall of the volume size, the mobile format conversion module **19** may make a decision to use the higher bitrate for the video content and the lower bitrate for the audio content, so as to make the best use of the available volume size.

[**0051**] If examination of the information stored in the production configuration data area **23** reveals that the target device is not optimised for video playback at the same frame rate as that of the DVD source **15**, then this is taken into account by the mobile format conversion module **19**. In particular, the mobile format conversion module **19** may modify the frame rate of the content data so that it is optimised for the target mobile device. Typically, this will involve a reduction in the frame rate which, because of the limited display size in most mobile telephones, would not be so noticeable as it would if a full size display were used. If the optimal frame rate is not equal to the source frame rate divided by an integer, then the mobile format conversion module **19** may use frame interleaving to effect a smooth result in the generated movie content when played back on a mobile telephone.

[**0052**] Step S**3** thus utilises information stored in the production configuration data area **23** to control the mobile format conversion module **19** to encode the data using the codec **22** into the appropriate data format and with appropriate bitrates.

[**0053**] The production configuration data area **23** may be updatable according to the target device which is of interest in a particular format conversion process. In this case, the production configuration data area **23** will store data for only one target device at a time, and this data is changed as required. Alternatively, the production configuration data area **23** stores a set of data for each of plural target devices, and one of the data sets is selected according to the particular target device of interest at a given time. In either case, the apparatus **10** is easily controlled to carry out a format conversion process which is optimised for each of plural target device configurations.

[**0054**] Digital rights management content is added to step S**4**. This is implemented by the mobile format conversion module **19** using the DRM processing module **21**. The procedure implemented by the step S**4** depends on the target format identified by the information stored in the production

configuration data area **23**. What form of DRM content is added may depend in particular on the form of the codec **22**. The form of the codec **22** in turn has an effect on the form of the codec in the media player. In particular, when the codec **22** is a custom codec, a custom form of DRM is used. Here, the form of DRM can be selected to provide optimal operation with the custom media player. If an off-the-shelf codec, such as Real Media™, is used as the codec **22**, a suitable DRM will be used.

[0055] Assuming it is allowed by the media player and the target device, the DRM content may impose content reproduction and distribution restrictions as follows. One option is to limit viewing of the content to the particular target device or user, as for example identified by an IMEI or an IMSI number or any other unique or quasi-unique serial number. In this case, the serial number needs to be included in the production configuration data area **23**, so that the mobile format conversion module **19** can operate with the DRM processing module **21** and the production configuration data area **23** to include suitable DRM content in the movie data. Another option is to allow the movie to be viewable up until a particular time and/or date. Thus, the resulting movie will have a "shelf-life" and will not be viewable after the date and/or time specified by the DRM content. A third option is to allow the movie content to be viewable on a predetermined number of occasions (N times). Once the movie has been viewed N times, the media player in the target device will not allow the content to be refused again, thereby rendering it useless. Alternatively, the media player may be arranged to erase the MMC or otherwise delete or corrupt the movie data immediately after the Nth viewing. Alternatively or in addition, the DRM content can prevent the content being copied or forwarded if not authorised. Thus, it can be said that the DRM content prevents or deters the consumption of the content on mobile devices other than the one for which it was intended and/or copying of the content.

[0056] Preferably, the DRM content is encrypted and included in the header of the resulting movie data, although the DRM content may be included in the movie data in any suitable way. Clearly, if a standard DRM process is required to be used by the target device, the DRM content included in the movie data by the mobile format conversion module **19** in the DRM processing module **21** will conform to the relevant standard.

[0057] The DRM processing module **21** also obfuscates the content at step S**4**. This is particularly important if the codec **22** is an industry standard codec, since otherwise it might be possible to render the content using a player other than one specifically designed for use with the content. Content obfuscation is performed by a command line-based obfuscation tool, forming part of the DRM processing module **21** as follows.

[0058] Content obfuscation is performed in frames. This helps to limit the overall CPU load when de-obfuscation is performed. The particular obfuscation method used depends on what format the content is in. For instance, the obfuscation method used with Real Media™ format content is different to that used with MP3 format content.

[0059] Only content data is obfuscated; content headers are not obfuscated. Optionally, only some of the content data frames are obfuscated. Of those content data frames that are obfuscated, some are obfuscated in full and some are only part obfuscated. For instance, key frames may be fully obfuscated, and only a portion (for instance the first ten bytes) of all other frames are obfuscated. Obfuscation involves making calculations such as bit shifting, adding and subtracting certain bytes depending on their position in the stream, etc. The calculations are based on the outcome of a suitable calculation on the timestamp in the content block header. The positions are derived by a modulo of the number of bytes in the content frame for every iterance of the calculation. The particular obfuscation technique used is not critical.

[0060] The obfuscation process is selected to as to require a relatively small, preferably minimum, CPU overhead for decoding by a media player when played back on a mobile device.

[0061] The first data content block is extended with an obfuscation identifier. This identifier is located at a suitable position in the content block, and the content header is adjusted to reflect the length of the content block including the obfuscation identifier. The obfuscation identifier is useable by a suitable media player to determine what obfuscation method was used by the DRM processing module **21**, which allows it to perform the corresponding de-obfuscation method. Following addition of the obfuscation identifier, it may be necessary to correct affected index entries so that seeking can be performed properly.

[0062] The obfuscation process used is different for different content batches. For example, the number of affected bytes can vary. Also, different compliments can be used, e.g. ones compliment in one batch and twos compliment in another batch. The obfuscation rules are configurable at content encoding. The obfuscation process used for a particular content batch can be selected randomly.

[0063] At step S**5**, the target content is written to the mobile format movie data area **20** as a file. The file may be an area of memory in a computer server, for instance, or the content file may be written directly onto an MMC or other portable transferable media. The file written by this step S**5** includes content in the appropriate format, and also DRM content either embedded into the movie content or else in a separate file. After step S**5**, the conversion is complete, the result is stored in the mobile format movie data area **20** data constituting the movie originally on the DVD data area **15** but encoded in a format suitable for use by the target mobile device and having appropriate audio and video content bitrates. Furthermore, the movie includes suitable DRM content, multiple volumes if appropriate to the format of the target device, a single audio sound track, and optionally a single subtitle track.

[0064] Where the video content on the DVD **15** has a different aspect ratio to the display of the target device, there preferably is modification of the video signal from the DVD such that it corresponds to the aspect ratio of the target device. This can be carried out by the DVD encryption and extraction module **18**. Preferably however, modification of the video signal from the DVD **15** such that it corresponds to the aspect ratio of the target device is carried out by the mobile format conversion module **19**. The modification may involve simple cropping from the left and right sides of images if narrower images are required, or cropping from the top and bottom of images if wider images are required.

The modification may involve as well or instead a limited amount of image stretching, either widthwise or heightwise. In this case, it is preferred to have more picture linearity in the central region of the display than at the edges of the display. Thus, compression or stretching is effected to a greater degree at the edges of the images than it is a central portion. The DVD encryption and extraction module **18** or the mobile format conversion module **19**, as the case may be, can be pre-programmed to make a decision as to what cropping and/or stretching is required on the basis of a look-up table relating course aspect ratios to target device aspect ratios and the corresponding modification required, or in any other suitable way.

[**0065**] In the embodiments described above, the data written to the mobile format movie data area **20** relates only to content data. In an advantageous embodiment, the data written to the mobile format movie data area **20** also includes one or more media players. This is advantageous for a number of reasons. Firstly, it reduces the number of factors which need to be taken into account by the mobile format conversion module **19**. The target device configuration information does not need to include information identifying the media player included in the mobile device, since this is not needed when the media player is included with the movie content data. Secondly, it allows movie content data to be consumed even if no suitable media player, or indeed no media player at all, is included in the mobile device.

[**0066**] The media player or players may be embedded, or alternatively included alongside, the movie content data. Embedding the media player into the content data allows easier control of the movie content, and makes it very difficult for the movie content data to be separated by unauthorised persons. Each media player typically consumes less than 1 MB of memory.

[**0067**] In one embodiment, a single custom media player is included with the movie content data. After the data is written onto an MMC card, the data relating to the media player is extracted by the mobile device from the MMC and the media player run to process the movie content data.

[**0068**] In another embodiment, a number of different media players are stored, along with the movie content data and a loader program. The mobile device is controlled to run the loader program initially. The program detects the relevant configuration of the mobile device and determines therefrom which of the media players to use to consume the movie content data. In this way, it is possible to utilise an MMC card for a greater number of target device configurations, which clearly can be advantageous, especially when the MMC cards are intended for retail from a shop display or similar.

[**0069**] If the media player is not a custom player, the loader program preferably is arranged to control the mobile device to detect whether or not it already includes a suitable media player. If a suitable media player is detected, this is controlled to be used instead of installing a media player from the MMC card onto the mobile device. This is advantageous since it reduces the possibility of there being an installation or deinstallation error, thereby improving the reliability of the mobile device.

[**0070**] Instead of including multiple separate media players, multiple media players may be provided through a single configurable media player software application. In this case, the loader program may determine what media player is required, and operate appropriate software modules forming part of the media player software. Software module or functions which are not appropriate for the mobile device configuration are not used. Thus, multiple media players are made up from a single software application, which reuses modules or functions for certain media player functionality. Where a single media player software application is used, the loader program may form part of the media player software application itself.

[**0071**] Any media player written to the mobile format movie data area **20** is able to render content, so includes suitable de-obfuscation functionality.

[**0072**] The movie content data, as well as any media player(s), stored in the mobile format movie data area **20** can be communicated to the target mobile device in any suitable way. For the next few years at least, it is envisaged that mostly MMC or other transferable media will be used to store and transport the movie content. However, as mobile data transfer becomes faster and cheaper, it is expected that movie content will be transferable over-the-air, for example using WAP or 3G data transfer. Transfer may instead be effected by transfer from an Internet connected PC which has downloaded the movie content from a website, using a short range link such as a cable, or wirelessly using IrDA or Bluetooth, or using a transferable storage medium such as an MMC.

[**0073**] A mobile device is shown schematically in FIG. **4**. Here, the mobile telephone **40** includes all the conventional components needed for voice communication, although these are not shown for the sake of clarity. The telephone **40** includes a movie decoder module **41**, which is in bidirectional communication with a codec **42**.

[**0074**] A movie is stored in a mobile movie data area **43**, which may take any suitable form. It may for example be an MMC, a memory space connected by way of an external drive, internal RAM or other memory, or it may take any other suitable form. A DRM validation module **44** is connected to receive DRM data from the data in the mobile movie data area **43**. The DRM validation module **44** controls the movie decoder module **41** to allow or disallow it to decode the movie data from the mobile movie data area **43** on the basis of a decision made using the DRM data, and time/date or serial number inputs as appropriate. When allowed by the DRM validation module **44** to decode movie data from the mobile movie data area **43** and when controlled to do so by user input, the movie decoder module **41** uses the codec **42** to decode the data and provide decoded data to a buffer **45**. From the buffer **45**, the movie is displayed on a display **46** by a display module **47**. The display module provides control data to the movie decoder module **41** so as to enable decoding at a suitable rate.

[**0075**] The mobile telephone **40** may be arranged to install a loader program from the mobile movie data area **43**, if one is stored there. The loader program then causes the mobile telephone **40** to determine its configuration, and to select a media player, which is a software application and which is also stored in the mobile movie data area **43**, accordingly. This media player then is used to consume the movie content data. If a suitable media player is already installed in the mobile telephone **40**, then this is used instead, and no media

player then is installed from the mobile movie data area **43**. However, using a proprietary media player stored in the mobile movie data area **43**, particularly although not exclusively in the case of the use of a portable storage device such as an MMC, can be advantageous since is allows effective control over the security of the content data, and allows other features not necessarily available with off-the-shelf or pre-installed media players.

[0076] The combination of an MMC and mobile device is illustrated in FIG. **5**. Here, the mobile device **40** is shown to include a CPU **51**, which provides video signals to the display **46**, via a display driver (not shown), and to an audio output device **52** (e.g. headphone socket or speaker, via an audio device driver (not shown). The CPU **51** is connected via a bus to ROM **53**, to RAM **54** and to an MMC connector and interface **55**. An MMC **56** is connected to the mobile device **40** by the MMC connector and interface **55**.

[0077] The MMC has stored in its internal non-volatile memory movie content data **57**, three different media players **58**, and a loader program **59**. When content is required to be played-out from the MMC, the mobile device loads the loader program **59**, which decides which of the media players **58** is most suitable by determining configuration parameters of the mobile device **40** and comparing them to parameters of the media players. This media player then is selected on the basis of the determination, is loaded onto the mobile device **40**, and is run (i.e. the media player program is processed) to reproduce the content from the content data **57**. As is conventional, operation of the media player **58** involves storing the media player program in the RAM **54**, and using the CPU **51** to extract relevant data from the MMC **56**, to decode it and to render the resulting content. The media player **58** removes the obfuscation identifier from the first data content block and adjusts the header, and uses de-obfuscation method to decode properly the content data. FIG. **5** is schematic, and detail not relevant to the invention is omitted.

[0078] The or each media player is arranged to detect the properties of the display **46** of the host mobile device **40**. In particular, the media player detects the display dimensions and orientation, in terms of numbers of pixels in height and width. The player is arranged to control reproduction of the video content on the display **46** in an orientation which is most suited to the mobile device **40**. If the display **46** is wider than it is high, then video content is reproduced with conventional orientation, i.e. without its orientation being modified. If however the display **46** is determined to be higher than it is wide, the media player reproduces the video content rotated by 90 degrees. Thus, the media player ensures that the video content always is reproduced in landscape format (wider than tall) regardless of screen dimensions. This allows more effective use of the area of the display **46**.

[0079] When the video content is rotated on a display **46** by the media player, the functions of a number of keys on a keypad (not shown) or other input device are caused by the media player **40** to be modified so as to be different to their functions when the video content is not rotated by the media player. Since the mobile device **40** will need to be rotated onto its side before the video can be viewed in its intended orientation, providing different key functions with different orientations allows the same control experience to be provided to a user regardless of the orientation of the mobile device **40**. Thus, modifying the controls allows control of the media player using the keypad or other input device to be more convenient and more intuitive for a user. The controls of particular importance are volume up/down, play, pause, forward and rewind, etc.

[0080] When the mobile device **40** is not a high specification device, i.e. it has relatively low content handling capability and/or a low resolution display, the media player is arranged such that it can access content from the MMC and not access content from other sources. This allows the content on the MMC to be optimised for reproduction by the proprietary media player, thus providing richer content reproduction than would otherwise be available considering memory size and other technical limitations of the MMC. This feature does not impinge on the ability of the media player to use a standard CODEC **42** pre-existing within the mobile device **40**. Indeed, the media player may utilise standard or other third-party CODECs, or it may utilise a proprietary CODEC.

[0081] When being run on higher specification mobile devices **40**, a different media player **58** is used. Here, the media player selected by the loader program **59** is one which is operable to scale non-optimal content for best presentation.

[0082] Alternatively, one media player **58** which has adjustable functionality is provided on the MMC **56**. Such a media player does not require a loader program. When running on a mobile device **40**, this media player **58** detects the relevant characteristics of the mobile device **40** and activates appropriate components and functionality of the media player **58** and refrains from activating other components and functionality.

[0083] The media player **58** includes a seek function. A user can move between chapters in content using the seek function. To allow this, the MMC is written with a placement file, in addition to the media file. The placement file has a file extension ".pm". It includes a line for each section or chapter. Each line comprises a section name, e.g. 'start of film', 'car chase scene', etc. Each line also includes a value which relates the timestamp corresponding to the start of that section. The timestamp and the section name are separated by a # character. When a key entry is made on a keypad of the mobile device **40** indicating that it is required to scan to the next section, the timestamp of currently played content is used to identify which line of the placement file relates to the next section. This involves determining the line that includes the smallest timestamp that is greater in value than the current timestamp. The timestamp from this line then is sent to the media player **58**. The media player **58** then starts playing content from that timestamp. Since this process is very quick to effect, it will normally have been implemented in less time than it takes for a user to make a second key entry. Thus, sections can be skipped quickly in succession. Sections can also be scanned through in reverse time order.

[0084] A digital fingerprint is included at various locations in the content data. The fingerprint for example can be 5 bytes long. The same fingerprint is placed at regular intervals throughout the content data. If obfuscation is used, each occurrence of the fingerprint also is obfuscated. The fingerprints are indistinguishable from the content, so the loca-

tions of the fingerprints cannot be determined by examining the data. Thus, a media player which decodes content without first removing the fingerprints will not get the correct data, and playback will fail.

[0085] Thus, the media player **58** needs to know what the fingerprint is and where the fingerprints are. This allows content playback to be effected only after authorisation by the server **80**. During an authorisation process, the server **80** informs the media player **58** what the fingerprint is and where it is found in the content data. Without knowing where fingerprint is, it cannot be removed from the content by the media player **58**. Also, the media player **58** ensures that fingerprint present in the content is as expected before it will play the content. The fingerprint is included in the first packet of the content data, so can be validated straight away.

[0086] The use of digital fingerprints allows some advantageous features.

[0087] For example, an MMC can be sold with one media item (e.g. a movie or TV show) unlocked for playback by the media player **58**. Other media items are provided on the MMC, but need unlocking before they can be played. In the menu of the media player **58**, the additional media items are shown as being locked. When one of these media items is selected, the media player **58** causes a media code to be displayed for that media items, and provides an entry window in which an unlock code can be entered.

[0088] To unlock the media item for playback, the user of the device **40** sends an SMS to the server **80** which includes the media code displayed by the media player **58**. The media code is 11 bytes (i.e. 11 alphanumeric characters) long. It is comprised of an obfuscated message including a media identification code, which identifies the corresponding media item, and the serial number of the media player **58**. On receiving the media code via SMS, the server **80** validates the code. Validation involves checking that the serial number relates to a media player **58** which exists, and checking that the media item exists. Once validated and once the media item is known to be paid for, the server **80** obtains a suitable unlock code, obfuscates it and sends the obfuscated unlock code to the device **40**. The unlock code includes the serial number of the media player **58** and the digital fingerprint which is included in the content data. Payment can occur through a WAP push SMS, which takes a WAP browser of the device **40** to a payment system such as Bango or mwallet. Following the server **80** receiving confirmation of payment from the payment system, the unlock code is sent by SMS from the server **80** to the device **40**. Alternatively, the server **80** could send a reverse billing SMS containing the unlock code, which the user is billed for by their mobile telephone service provider. Alternatively, this could be done automatically using an http connection, on selection of a "buy" option by a user.

[0089] On receiving the SMS, the user enters the received obfuscated unlock code included in it into the entry window provided by the media player **58**. The media player **58** then de-obfuscates the unlock code, and checks that the serial number forming part of the unlock code matches the serial number of the media player **58**. A suitable message is displayed if there is not a match, since the user may have entered the unlock code wrongly. If there is a match, the media player writes the unlock code, including the digital fingerprint included in, it into the MMC, to unlock the media item. To play the media item, the user then accesses it through the menu provided by the media player **58**. The media player **58** then accesses the unlock code from the MMC, which allows the media item to be played if the fingerprint in the unlock code is the same as that included within the content data.

[0090] If the fingerprint in an unlock code does not match the fingerprint present in the content, the media player **58** is arranged to delete any unlock code for that media item from the MMC, and revert to the menu. This allows operation of the media player **58**, and the possibility of entering the correct unlock code, even if the unlock code entered into the MMC is wrong, for example because it was entered in respect of the wrong media item.

[0091] An optional additional feature ties the MMC to a particular device using the IMEI of the device. When first installed on a device and before registration, the media player **58** knows the serial number of the content, and knows the media identification code, but does not know the digital fingerprint. The fingerprint is needed before the content can be played. The media player **58** initiates an http connection to the server **80**. The media player **58** then registers by submitting the serial number of the media player **58** and the IMEI number of the device **40** to the server **80**. The server **80** looks-up the content identifier on the basis of the received information, and stores the received information. The server **80** then sends an encrypted message comprising the IMEI, the serial number of the media player **58**, the fingerprint and information identifying the locations of the fingerprint in the content. The media player **58** stores the encrypted message on the MMC. The media player **58** then is able to play the content. If the MMC then is moved to another device **40**, which necessarily will have a different IMEI, it will not be played by the media player **58**. In particular, when the media item is selected to be played, the media player **58** checks the serial number, the media identification code and the IMEI forming part of the encrypted message stored on the MMC. The media player **58** is arranged such that if the IMEI stored as part of the encrypted message does not match the IMEI of the device **40**, the media player **58** will not play the content. To enable the content to be played on a different device **40**, the content must first be unregistered from the original device. It may then be registered on the new device **40** in the manner described above. Additional media items on the MMC can be purchased in the same way as that outlined above. Furthermore, different media items may validly be registered onto different devices, as long as each media item is registered only onto a single device **40**.

[0092] If content stored on an MMC is allowed to be copied freely by users, this technique can be used to track copies. Here, deregistration is not required. An MMC can be registered successively onto different devices **40**. Furthermore, there can be an unknown number of copies in existence. When an MMC is loaded onto a device which has not been registered for the content, the media player **58** contacts the server **80**, which registers the content. Following registration, the server **80** sends to the device **40** the encrypted message which allows it to play the content. By monitoring registrations, the server **80** can determine how many copies of the MMC are made. The server **80** can also determine an approximate geographical distribution of them by detecting the gateway IP address of the network element through which the content is registered.

[0093] The hardware of the MMC **56** may be standard, for example any of the MMC forms which currently are publicly available. A typical MMC hardware design consists of a flash memory device and a memory/interface controller residing on a very thin PCB (printed circuit board) in a very low profile plastic housing. The underside of the PCB generally forms the bottom of the housing. There are a number of different sizes of MMC.

[0094] According to certain aspects of the invention, the MMC hardware is non-conventional, and includes additional security features. In this case, a proprietary media player **58** is used to unlock and read content on the secure MMC.

[0095] A first embodiment of a novel MMC will now be described with reference to FIG. **6**. Here, an MMC **56** includes a housing **60** in which connector pins **61** are provided. The connector pins form part of a host communications interface to an external device, such as the mobile device **40**. The MMC **56** also includes non-volatile memory **62**, connected to a memory and interface controller **63**, which controls access to the memory **62** and interfaces to the connector pins **61**. The MMC thus far described is conventional. The MMC **56** also includes a security device **64**, which is not conventional. The security device **64** is interposed between the memory and interface controller **63** and the connector pins **61**. Thus, the memory and interface controller **63** and the data (DAT), command (CMD) and clock (CLK) ones of the connector pins **61** are not connected directly since at least some connection between these components is via the security device **64**. VCC, VSS1 and VSS2 ones of the connector pins **61** are connected to both the security device **64** and the memory and interface controller **63** in parallel. The security device **64** may be implemented as a microcontroller, an ASIC (application specific integrated circuit) or an FPGA (field programmable gate array). The components of the MMC **56** are mounted onto a PCB (printed circuit board), which forms part of the housing **60**. Thus, the MMC **56** may have the same dimensions and the same external connectors as a conventional MMC.

[0096] The security device **64** is arranged to intercept data and commands communicated between the host device, e.g. the mobile device **40**, and the memory and interface controller **63**. This intercepted data is processed and either is passed through the security device **64** modified or unmodified, or alternatively is replaced by data generated by the security device **64** itself.

[0097] Specific data or commands passed in any response can switch the security device **64** into an active mode, in which the security device **64** reads or writes to one of the memory and interface controller **63** and the host interface **61**, masquerading as the other one of those devices. In the active mode, the security device **64** also independently, i.e. without external control, interrogates the memory and interface controller **63** and either prepares data for subsequent host requests or writes data to the non-volatile memory **62** for subsequent requests.

[0098] The provision of the active mode allows copy protection to be achieved through cooperation between the MMC **56** and the media player **58**.

[0099] The security device **64** does not restrict access to regions of the non-volatile memory **62** where unprotected content resides, in both read and write modes. This allows the MMC **56** including the security device **64** to be used conventionally, i.e. without the security features provided by the security device being operational. The security device **64** can be activated only by authorised entities, such as those licensed to place copyright content, e.g. movies, onto the MMC **56**.

[0100] The MMC **56** and the media player **58** are provided with the same serial number. During configuration, the media player **58** is provided also with the result of application of the derail number to a hash function, hereafter termed the hash of the serial number. The memory and interface controller **63** is controlled by the security device **64** to store at programming time (i.e. when it is programmed before sale) the serialised data serial number, a preconfigured security code, and the hash of the serial number.

[0101] Validation of the MMC **56** by the media player **58** and validation of the media player **58** by the MMC **56** will now be described with reference to FIG. **8**.

[0102] When the MMC **56** with content, one or more media players **58** and optionally a loader program **59** loaded onto it is connected with a mobile device **40**, the media player is made visible in a menu thereof, and thus becomes able to be activated as with any other software application present on the mobile device **40**. When the media player **58** first is started, a first security validation is implemented, in which the following occurs. Firstly, the most suitable media player **58** is uploaded to the mobile device **58**. The media player **58** then at step S**8**.**1** sends the hash of the serial number to the security device **64**. The security device **64** at step S**8**.**2** then compares this with its internally stored hash of the serial number. If the comparison at step S**8**.**3** reveals a match, it is initially assumed that the media player **58** and the MMC **56** are matched, and the security device **64** unlocks the MMC **56** at step S**8**.**4**. The security device **64** then sends at step S**8**.**5** the preconfigured validation code to the media player **58**. Alternatively, if the comparison does not reveal a match, the security device **64** at step S**8**.**6** does not respond. When the media player **58** receives a validation code, it performs at step S**8**.**7** a 32 bit CRC (cyclic redundancy check) calculation on the validation code. On the basis of this calculation, the media player **58** determines at step S**8**.**8** whether the MMC **56** is the one associated with the media player **58**, and unlocks the media player at step S**8**.**9** if appropriate, or else aborts with an error message at step S**8**.**10**. At this stage, the media player **58** can read data from unprotected areas on thereon-volatile memory **62**, if any such areas are present.

[0103] A second stage security check is performed when playing the content. After the media controls on the MMC **56** are unlocked and the data becomes readable, data is read out from the non-volatile memory **62** to the media player **58**. In parallel with this, the data stream is set at step S**8**.**11** into frames of 1 kB, i.e. there are 1000 bytes between frame start and end points. The media player at step S**8**.**12** calculates the security code (as described in more detail below) and then sends it to the security device **64**, where it is decoded at step S**8**.**13**. On the basis of the decoding, the security device **64** determines at step S**8**.**14** if the security code is valid. If invalid, the security device **64** at step S**8**.**15** resets a timeout counter, thereby preventing a timeout occurring and locking the content. If valid, the memory and interface controller **63**

at step S8.16 considers the subsequent data frame as being validated for access. If a valid code is not received before the end of this frame, subsequent frames are filled with random data instead of content data.

[0104] The media player **58** recalculates the correct security code once in every frame, but generates 20 security codes for each data frame, 19 of which are incorrect. The media player **58** sends the MMC **56** all the security codes at step S8.17, in this example resulting in 20 security codes being sent for every frame of data. 19 of these codes are intentionally incorrect, and only one of them is correct. The security device **64** of the MMC **56** compares the results of its calculations with the security code sent by the media player **58**. The security device **64** allows content data to be sent to the player as long as one correct security code is received in every frame. If the security device **64** detects that a valid security code has not been received for a predetermined period of time, using a timer, or if too few codes (either correct or incorrect) are received, then the security device **64** disables access to the data in the non-volatile memory **62**. The security device **64** then needs to be unlocked again by the media player **58** before content playback can be resumed. The security device **64** also locks the MMC **56** if it has not been accessed for a predetermined, configurable period of time.

[0105] The security code is calculated based on the following data:

CRC	the last 4 bytes of the decoded validation code (the checksum part)
Bytes	the total number of bytes read from the MMC 56 so far
Random	a number between one half of the number of security updates per frame (in this case, 10 is half of the 20 updates per frame that there are) and 0.

[0106] The media player **58** performs the calculation:

((*CRC*<<Mod 32(Bytes))Xor(Bytes))*Random

[0107] This means that the checksum part (CRC) of the validation code is shifted left by a modulo of 32 of the number of bytes read. The result is Xor-ed with the number of bytes read. The Xor operation consists of applying corresponding bits in the two numbers to respective exclusive-or gates. The result is multiplied by the random number Random.

[0108] The security device **64** in the MMC **56** performs the calculation:

((*CRC*<<Mod 32(Bytes))Xor(Bytes))*Moduloframe size(frame number)

[0109] Moduloframe size(frame number) is frame number modulo 1000 in this instance because the frame size may change, i.e. 1000 e.g. 5032 becomes 0032.

[0110] The result of this is the continual validation of the media player **58** by the security device **64** of the MMC **56**. This prevents it being possible to use a false media player to extract the content data in a useable form; instead the data can only be extracted from the MMC **56** by the correct media player **58**, which renders the content for consumption but does not allow the content data to be used to provide unauthorised copies. The fact that the media player **58** sends many incorrect codes makes it difficult or impossible to determine from examination of the codes sent from the media player **58** to the MMC **56** what calculation is needed to determine the correct codes, thus increasing security since the difficulty of making a false media player which could extract data from the MMC is significantly increased.

[0111] Using these features, the security device **64** is operable to determine whether an external device, comprising the mobile device **40** running the media player **58**, is entitled to access content data from the non-volatile memory **62**, and to allow or disallow access to the content data accordingly.

[0112] An alternative MMC **70** is shown in FIG. **7**. Here, the memory and interface controller **63** is omitted. Instead, a combined memory and interface controller and security device **71** connects the non-volatile memory **62** with the connector pins **61**. This provides the same functionality that the memory and interface controller **63** and the security device **64** do together, but with some additional functionality, as explained below. This embodiment has an advantage in that it could be included within a smaller housing than the FIG. **6** MMC **56**. Since it has less hardware, it may also be less expensive to manufacture. Also, the combined memory and interface controller and security device **71** does not need to support the same type of non-volatile memory as a MMC controller, thereby providing component flexibility.

[0113] The combined memory and interface controller and security device **71** emulates the host interface of a standard MMC controller, so as to allow full connectivity with host devices, such as the mobile device **40**. It also supports additional host interface commands to support security configuration and security validation in some specific hosts. The combined memory and interface controller and security device **71** encrypts all data written to the non-volatile memory **62**, and decrypts all data read from the non-volatile memory **62**. Thus, data accessed by the mobile device **40** is not read from the non-volatile memory **62** directly; instead it is decrypted, processed and buffered in the combined memory and interface controller and security device **71**.

[0114] Some data accessed by the host is a result of processing, for example the security device **64** compiles information for subsequent host requests, or is status information, e.g. security status information, which the media player **58** can use to re-validate security or inform the user of the nature of a problem

[0115] The combined memory and interface controller and security device **71** can be implemented by a microcontroller, an ASIC or an FPGA.

[0116] With the MMCs of both FIGS. **5** and **6**, DRM information is stored in a DRM file within an area of the non-volatile memory **62** which has been defined as a secure area during MMC configuration. The media player **58** can read the DRM file but not influence it, except in the case of a time specific DRM matter. The security device **64** or **71** is arranged to count the number of times that the content is played. If the content is only partially played, this is counted as a play of the content. The number of times that the content has been played is recorded in the DRM file by the security device **64** or **71**. This information can be read by but cannot be not influenced by the media player **58**. The DRM file indicates a maximum number of occasions in which the content data can be played out.

[0117] The DRM data also includes a timeout date or validity date for the content. When the media player **58** is first started, it cooperates with the security device **64** or **71** to write the current time and date of the mobile device **40** from its internal clock (not shown) into the DRM file. If playback of the content is requested and the security device **64** or **71** determines that the latest time and date at which the content could be played has expired (i.e. the current time and date is later than the time/date first recorded plus the validity period), the security validation between the security device **64** or **71** and the media player **58** fails, and an appropriate message is delivered to the user via the display **46**. The same occurs if the limit of the number of occasions on which the content data can be played out is reached. The security device **64** also writes to the DRM file data identifying that the content has expired.

[0118] If after the content has expired once and the time/date of the mobile device **40** is changed to a value that precedes the expiry time/date of the content, the security device **64** or **71** can detect this by detecting data identifying that the content has previously expired in the DRM file. In this case, a predetermined number of further plays of the content, for example 5 plays, are allowed before the content becomes locked requiring a DRM unlock. This is achieved using on-line validation. This feature eases the user impact if the clock in the mobile device was incorrectly configured when the media player **58** was first started.

[0119] The on-line validation process commences with the media player **58** connecting to a DRM server, shown at **80** in FIG. **5**, for example belonging to an entity that is licensed to render content onto MMCs. The DRM server **80** knows the configuration of every MMC **56** that has been released. Connection may be made through WAP or SMS, or in any other suitable way. If the DRM information on the DRM server is valid, the DRM server sends a code through the media player **58** to the security device **64** or **71**, which causes it to be validated and thus unlocks the content for further playback. This involves updating the DRM file. Locked content can be unlocked again by payment for further content access through a variety of channels (web, wap (e.g. Bango) and SMS).

[0120] If content on an MMC **56**, **71** is locked, the media player **58** will not play the content data back. In this case, the user of the mobile device **40** may arrange for the content to be unlocked for further playback, for example by making an additional payment. This can occur in any suitable way, for example using WAP, a web-based payment service, or by negotiating with an operator by telephone. When payment is made, the DRM server **80** is updated with this information. When the user subsequently starts the media player **58** and attempts to access the locked content, the media player **58** contacts the DRM server **80**, in any suitable way, which sends an unlocking code to the mobile device **40** which the media player **58** passes to the security device **64** or **71**. The security device **64** or **71** then validates the unlocking code, and updates the DRM file to unlock the content. This may involve the use of digital fingerprints, as described above.

[0121] If an MMC **56** such as one of the FIGS. **6** and **7** MMCs **60**, **70** is used, some high-level protection is possible through suitable design of the MMC hardware. Circumvention of such protection would require reverse engineering of the media player **58**.

[0122] Broadly speaking, certain sectors of the MMC **56** are arranged not to contain correct data (and hence not correct "next sector" data) until specific data has been read and written from other, specific, sectors on the MMC **56**. Some of the media player **58** system files and part of the content will reside over some of these sectors. This copy protection implementation consists of custom MMC hardware implementation, MMC build tool functionality (special format etc), and support by the media player **58** (which must write and read the right security sectors).

[0123] At start-up of the media player **58** and at regular intervals thereafter, the media player **58** writes specific data to some unused sectors on the MMC **56**. These sectors are unused because the format of the file system on the MMC **56** (as specified by the boot sector) does not include these sectors. The data written to these sectors is processed by the MMC **56**, and the result is reflected in a number of file sectors. The result is the correct data for the sector.

[0124] In particular, specific data areas on the MMC **56** contain data that is a result of an algorithm applied to data written to other areas of the MMC **56**. For instance, this could be a combined hash function of 10 writes to another block of a smaller size. E.g. from 15360 to 15871 bytes on the MMC **56** is calculated by a hash generated from data written to bytes **15360** to **15871** over 10 consecutive correct writes. This calculation places expected content in bytes **12800** to **13311**. This block will be a sector in a file that the media player **58** is about to read from. Importantly, the next sector in the file is referenced from this sector. The media player **58** is arranged such that, if this sector does not contain correct data when accessed by the media player **58**, the media player will shutdown (and may even crash).

[0125] Some of these writes generate garbage content in the read area, and some of the writes generate genuine content that the media player **58** will use. Because the media player **58** writes many times, most of the time with bogus information, deciphering of this process is time consuming and a good deterrent against copying. Thus, this is effective in significantly hindering reverse engineering techniques.

[0126] An example of this is as follows. Three sectors on the device, each of 512 bytes, are target sectors that are referenced as part of a file on the MMC. These target sectors contain the results of calculations on the data contained in source sectors. Nine source sectors, each of 512 bytes are used.

[0127] Accordingly, three source sectors are processed per target sector. An example calculation is as follows: the first target sector is contains all of the bytes of source (unsigned calculation) sector **1** (1^{st} byte to last)+source sector **2** (1^{st} byte to last) xor source sector **3** (last byte to 1^{st})

[0128] Source sector **1** resides on block **12034**, **2** on **12044**, **3** on **12054**, **4** on **12035**, **5** on **12045**, **6** on **12055**, **7** on **12036**, **8** on **12046**, **9** on **12056**

[0129] Target sector **1** is associated with source sectors **1**, **2** and **3**, and resides on sector **8000**. Target sector **2** is associated with source sectors **4**, **5** and **6**, and resides on sector **8001**. Target sector **3** is associated with source sectors **7**, **8** and **9**, and resides on sector **8002**.

[0130] The resource file for the media player **58** resides across target sectors **1**, **2** and **3**, and content resides over 5,

6, 7, 8 and 9. Reading these files without writing the correct information into the source sectors will result in file corruption.

[0131] Although the mobile device **40** is said to be a mobile (cellular) telephone, it may instead be a personal digital assistant (PDA), which may or may not have bidirectional voice communication capabilities. The invention is primarily concerned with providing audio-visual content on a device which is designed primarily for another function. However, the invention is concerned also with dedicated media players.

[0132] Also, although certain aspects of the invention have been described in relation to an MMC **56**, this is not essential. Instead of an MMC, other type of medium including non-volatile memory and an internal memory controller with access to content data stored on the memory being obtained through an interface could be used instead. For example, a memory device with a USB or Bluetooth™ or other interface could be used instead. The housing of the memory device may take any suitable form.

[0133] An alternative scheme for protecting the content stored on the MMC using digital fingerprints and obfuscation will now be described with reference to FIGS. **9**, **10**A, **10**B, **10**C and **10**D.

[0134] FIG. **9** illustrates a sequence **1200** of data packets stored on the MMC. First in the sequence **1200** are first to Zth headers, of which the first, second and Zth headers **1201**, **1202**, **1203** are shown. Immediately after the Zth header **1203** is a first content block **1204**, which is the first of Z content blocks in sequence. The first, second, Xth, Yth and Zth content blocks **1204**, **1205**, **1206**, **1207**, **1208** are shown in the Figure. The headers each have a one-to-one relationship with a respective content block, and are in the same sequence, i.e. the Sth header relates to the Sth content block.

[0135] Following the Zth content block **1208** is an index section, which comprises first to Jth index headers and first to Jth index entries. A 1st index header **1209** if followed by a 1st index entry **1210**, which is followed by a second index header **1211** etc. An index header thus is immediately followed by the index entry to which it relates. The final data is a Jth index entry **1212**. Typically, J is a number much smaller than Z.

[0136] Bach of the content blocks **1204-1208** relates either to audio or video content. Each content block includes a timestamp, so that a media player can relate audio content to the corresponding video content. The header of a content block **1204-1208** includes information which identifies which stream the content block relates to. In most cases, there will be only one audio stream and only one video stream, although this may vary.

[0137] In this example, the content blocks are produced by a RealProducer tool, so the headers **1201-1203** and the data in the content blocks **1204-1208** are compliant with the Real format. The content is decodable by a Real codec. Real is a trademark of RealNetworks, Inc. With Real, each content block **12045-1208** which relates to a video stream contains data relating to a whole video frame. Thus, each content block includes all the information relating to a frame, and relates to only one frame. A key frame is provided periodically, and each non-key frame between successive key frames detail differences between the previous frame and that frame. The interval between successive key frames is a settable parameter in RealProducer. The size of content blocks is variable. Where there is a lot of difference between two successive frames, the content block for the second frame includes a relatively large amount of data. Where there is less difference, the second frame typically contains less data. Where there is no difference between successive frames, the second content block may contain only a few bytes of data. Each header **1201-1203** includes information which identifies whether or not the corresponding content block **1204-1208** includes data of a key frame.

[0138] It is not possible for a machine or human operator to determine from examining the data of the content blocks **1204-1208** where the boundaries between the content blocks lie. However, each of the content headers **1201-1203** includes data which indicates the length in bytes of the corresponding content block **1204-1208**. Thus, the start address of a content block **1204-1208** can be determined by adding the start address of the previous content block the length of the previous content block and one additional byte. Adding the length of the previous block to the start address of that block results in the address of the final byte of that block, thus the start address of the following content block is found by adding an additional byte to that address.

[0139] Thus, if any bits or bytes are added or taken away from any of the content blocks **1204-1208**, the actual start addresses of all following content blocks will not match the addresses calculable from the headers **1201-1203**. In this event, the data fed into a Real decoder will not be decoded properly, and the media player would probably stop working altogether and require closing and reopening before it could become operational.

[0140] The inventors exploit this to advantage. In particular, some content blocks **1204-1208** are each provided with one or more excess data items. The excess data item preferably is a digital fingerprint. In the Figure, the 2nd, Xth and Yth content blocks **1205**, **1206**, **1207** are provided with a digital fingerprint.

[0141] It clearly is important for a media player which is to playback the content to know where the excess data items are. Otherwise, data sent to a decoder in the media player may not be in the correct format, and thus will not be decoded properly and may crash the media player.

[0142] There are numerous ways in which a media player can be informed of the locations of excess data items in content data. The inventors have found that a particularly good solution is to utilise a pre-determined scheme to determine where to include excess data items before recording the data onto the MMC, and to program the media player with details of that scheme.

[0143] In some embodiments, excess data items are included in the second content block **1202** and at regular intervals (in a playback or presentation time sense) thereafter. For instance, excess data items can be included in the first content block having a timestamp immediately following an integer multiple of a predetermined excess data item interval, for instance 20 seconds. According to this technique, excess data items are included in the first content block having a presentation time after 20 seconds, in the first content block having a presentation time after 40 seconds, the first content block having a presentation time after 60 seconds and so on.

[0144] The location of the excess data within the content block **1205-1207** also is important. In some embodiments, the excess data items are included at the same position in each of the content blocks **1205-1207** in which it present. For instance, the excess data may be included from. byte **56** of those content blocks **1205-1207**. Bytes **57** onwards of the content block are retained, but after the digital fingerprint. If a content block includes 56 content blocks or fewer, then the excess data items are included at the end of that content block, after all of the content data. Thus, if a content block includes 20 bytes, then the excess data begins at the 21st byte.

[0145] There are numerous ways in which a media player can be informed of the locations of excess data items in content data. The inventors have found that a particularly good solution is to utilise a pre-determined scheme to determine where to include excess data items before streaming the data, and to program the media player with details of that scheme.

[0146] For instance, there may be twenty different schemes that the media player is programmed to handle. Each scheme is identified by a different digital fingerprint scheme identifier. This may form part of the media code, may be included at a suitable location in the content, similarly to the obfuscation identifier, or may be identified to the media player in any other suitable way. A first scheme may involve digital fingerprints included at byte **56** in each of the first content blocks following integer multiples of 20 seconds, as described above. A second scheme may involve including a digital fingerprint at byte **32** of every content block immediately following integer multiples of 30 seconds. A third scheme may involve including a digital fingerprint at byte **32** of the content block six content blocks following integer multiples of 30 seconds. A fourth scheme may involve including a digital fingerprint at byte **32** of every content block immediately following integer multiples of 60 seconds and at byte **11** of every content block immediately following integer multiples of 30 seconds unless there is a digital fingerprint at byte **32** thereof, i.e. the location of the digital fingerprint alternates between byte **11** and byte **32** between successive occurrences of it. In another scheme, the content block in which the fingerprint is located is varied. For instance, the digital fingerprint may be located relative to a content block immediately following integer multiples of 40 seconds, at three content blocks following, seven content blocks following and two content blocks following in a repeated sequence. The more the location of the digital fingerprint is varied, the greater the protection that is afforded.

[0147] The length of the excess data items are not critical, although to avoid increasing the size of the resulting data by a significant degree the excess data preferably is not unduly long.

[0148] Since the excess data items are intended to be removed before decoding, the form (i.e. content) of the excess data items are not necessarily important. However, the inventors prefer that the excess data items are in the form of a digital fingerprint. Preferably, each occurrence of the digital fingerprint is the same, i.e. has the same data sequence. For instance, the digital fingerprint may be 5 bytes long. Even if a third party manages to determine the data constituting the digital fingerprint, data strings having the same data will be present at numerous locations in the content data, so this information alone would not be enough to allow the digital fingerprints to be removed.

[0149] A content block, for instance the Xth content block **1206**, is shown in FIG. **10**A. The content block **1206** includes m bytes of data. In FIG. **10**B, the content block **1206** is shown with a digital fingerprint **1301** added, and is labelled **1304**. The digital fingerprint **1301** separates the m data bytes into two sections **1302**, **1303**. The first section **1302** includes data bytes **0** to n, and the second section **1303** includes data bytes n+1 to m. The length of the content block with the fingerprint is equal to m plus k bytes, where k is the size of the digital fingerprint. Using the example given above, n is 56 and k is 5.

[0150] To provide additional protection against unauthorised playback and/or copying, some or all of the content blocks are obfuscated before they are stored on the MMC. Where a content block includes an excess data item, such as a digital fingerprint **1301**, then the excess data item is obfuscated along with the data forming the original content block. Simply, obfuscation comprises altering the data so that the resulting obfuscated data is different to the original data and cannot be decoded properly without first being deobfuscated. Obfuscation typically does not alter the amount of data, so the size of a content block is the same before and after obfuscation. Obfuscation is discussed above in relation to FIG. **3**. The content block **1304** including the digital fingerprint **1301** is shown obfuscated at **1305** in FIG. **10**C. The obfuscated content block **1305** includes k plus m bytes, as with the digitally fingerprinted content block **1304**.

[0151] The first content block **1204** is extended with an obfuscation identifier **1306**, as shown in FIG. **10**D. The obfuscation identifier **1306** is located at a suitable position in the first content block **1204**, and the corresponding content header **1201** is adjusted to reflect the length of the content block including the obfuscation identifier **1306**. The obfuscation identifier **1306** identifies what obfuscation method was used by the DRM processing module **21** to obfuscate the data. This allows a suitable media player it to perform the corresponding de-obfuscation method. Following addition of the obfuscation identifier **1306**, it may be necessary to correct affected index entries so that seeking can be performed properly. The first content block **1204** is not obfuscated, since otherwise the obfuscation identifier **1306** would be obfuscated.

[0152] If only one obfuscation scheme is used with all content, then the media player **58** knows what obfuscation is used, and thus the obfuscation identifier **1306** can be omitted from the first content block **1204**.

[0153] Since the content blocks **1203-1208** which include a digital fingerprint include more data than is indicated by their corresponding header **1201-1203**, the index entries **1210**, **1212** would be incorrect. In particular, the physical addresses pointed to by data forming part of the index entries **1210**, **1212** are supposed to point to the address of the beginning of key frames. However, if a digital fingerprint is included in one or more content blocks preceding that key frame, the actual address of the beginning of the content block containing key frame data will be higher than the address indicated by the corresponding index entry **1210**, **1212**. Accordingly, the data in the index entries **1210**, **1212** is modified to reflect correctly the actual start addresses of

the content blocks that the data is intended to correspond to. Thus, when the data of the index entries **1210**, **1212** is used by the media player **58** to access content, the content is decoded and played back correctly.

[**0154**] Following obfuscation of some or all of the content blocks **1205-1208**, the provision of a suitable obfuscation identifier **1306** in the first content block **1204** and the modification of the data in the index entries **1210**, **1212**, the resulting data is stored on the MMC **56**.

[**0155**] When the MMC **56** is inserted into a mobile device **40**, the mobile device **40** loads the media player **58**. When an item of content is selected for playback by a user through a menu of the media player **58**, the media player **58** begins to read the headers and the content blocks relating to that content. The DRM validation module **44** within the media player **58** knows the location of the obfuscation identifier **1306** within the first content block **1204**, and extracts it. The DRM validation module **44** of the media player **58** then uses the obfuscation identifier **1306** to determine what obfuscation method is needed to de-obfuscate the content data **57**. The DRM validation module **44** also determines whether the media code needed to playback the content data **57** is present. During an authorisation process, the server **80** provides the media code to the media player **58**. The media code also identifies the digital fingerprint **1301**, and allows the media player **58** to determine in which content blocks **1205-1208** and where in those content blocks the digital fingerprint is present. This can occur in any suitable way. For instance, the media code may include a digital fingerprint location code, which identifies a predetermined scheme useable to remove the occurrences of the digital fingerprint. The media player **58** then is ready to playback the content. However, the media player **58** preferably is arranged to playback the content only if the digital fingerprint included in the content blocks is the same as the digital fingerprint included in the media code. This provides a further check that the user is entitled to playback the content using the media player **58**. The fingerprint is included in the second content block **1205**, so can be validated straight away after playback of the content is commenced.

[**0156**] Without knowing where fingerprint is, it cannot be removed from the content by the media player **58**. Also, the media player **58** ensures that fingerprint present in the content is as expected before it will play the content.

[**0157**] In playing back the content, the DRM validation module **44** de-obfuscates those content blocks which are obfuscated. For instance, in respect of the content block **1305** of FIG. **10**C, the DRM validation module **44** performs the inverse of the obfuscation performed at the DRM processing module **21**, thereby obtaining the fingerprinted content block **1304**. In playing back the content, the DRM validation module **44** also removes fingerprints from the content blocks that include them. For instance, in respect of the content block **1304** of FIG. **10**B, the DRM validation module **44** removes the digital fingerprint **1301**, thereby obtaining the content block **1206** shown at FIG. **10**A. The content blocks are fed to the codec **42** of the media player **58** only after de-obfuscation and after removal of the digital fingerprints. Failure to do either of these actions would result in incorrect data being fed to the codec **42**, likely resulting in the crashing of the media player **58**. At best, content would not be played back in a useable form.

[**0158**] If a media player other than the media player **58** is used to attempt to playback the content, it will fail. In order to construct a media extractor which could extract useable content, the media extractor would need to know exactly what obfuscation method to use, and exactly where in the content blocks the excess data items are and what size they are. Thus, this technique provides substantial protection against unauthorised use of the content. Even with a relatively simple obfuscation method and relatively infrequent digital fingerprint inclusion, the protection afforded is such that it is technically more straightforward to extract content from an encrypted DVD than it is to take content from the MMC **56**. Since the media player **58** knows what de-obfuscation method is used and from what locations the digital fingerprint needs to be removed, it can playback the content correctly. However, the media player is not a media extractor, so cannot be used to extract the content in unprotected form for unauthorised use.

[**0159**] The use of digital fingerprints and obfuscation is useful in protecting transmitted content as well as content stored on a carrier. An overview of a broadcast television and radio system will now be described with reference to FIGS. **12** and **13**. Referring to FIG. **11**, a broadcast server **100** includes an input **101** at which channel feeds are received, and input/outputs to the Internet **102**. A mobile terminal **103**, for instance the mobile terminal **40**, is connected to the Internet **102** through a mobile network (not shown), and thus is able to communicate with the broadcast server **100**. Briefly, once a user of the mobile terminal **103** has subscribed to broadcast services, they are able to request streaming to them of data through which they can view a television channel or listen to a radio channel using the mobile terminal **103**.

[**0160**] The broadcast server **100** includes a WAP registration module **104**, through which a user of the mobile terminal **103** can become registered with the broadcast server **100** through a WAP connection to the Internet **102**. The mobile terminal **103** may be identified for example by its IMEI number. The registration module is in two-way communication with a registration database **105**, which maintains details of registered users and which allows a supervisor to monitor registered users and to unregister them as required. Following registration, the user is able to subscribe to services using a WAP connection between a billing module **106** and the Internet **102**. The billing module **106** is in two-way communication with a billing database **107**, which monitors subscriptions and allows a supervisor to examine individual subscriptions and to provide subscription statistics. The billing database **107** and the registration database **105** are in two-way communication with one another, so that registration information can be passed to the billing module **106** and subscription and billing information can be passed to the registration database **105** and the registration module **104**.

[**0161**] A channel configuration database **108** maintains configuration parameters of channels between the broadcast server **100** and multiple mobile terminals, only one of which is shown at **103** in the Figure. Channel configurations are passed from the configuration database to a channel configurator **109**, which has an http connection to the Internet **102** The channel configuration database **108** contains configuration data for the channels. The channel configuration database **108** is updated using a web based administration

tool to add, modify and remove channels to conform to incoming streams, which are setup by a manual configuration process.

[0162] The data included in the channel configuration database **108** consists of the full URL of each channel.

[0163] The channel configurator **109** reads the channel configuration database **108**, prepares an XML list of all channels available to a particular user (i.e. the channels to which they have subscribed) and sends this XML list to the media player **58** on the mobile device. A menu option to "refresh channels" within the media player can be used to initiate this process. The media player then creates a new channel list for the user.

[0164] Data received at the channel feeds input is processed by a chain of components comprising stream converters **110**, stream buffers **111**, a content encoder **112**, a DRM encoder **113** and a content server **114**. In this example, the streamed content is in Real™ format, so the content encoder **112** is RealProducer M and the content server **114** is RealServer™, although any other suitable format may be used instead. There is a stream converter **110** and a steam buffer **111** for each incoming channel feed at the input **101**. The stream converters **110**, the content encoder **112** and the DRM encoder **113** receive channel configuration information from the channel configuration database. The DRM encoder **113** also receives subscription information from the billing database **107**.

[0165] The content server **114** supplies streamed channels to the Internet **102**, from where they can be accessed by the mobile terminal **103** by applying a stream request to the content server **114** via the Internet **102**.

[0166] Using the broadcast server **100**, the user of the mobile terminal **103** is able to register and subscribe to services. When subscribed, the user of the mobile terminal **104** is able to select a channel from the content server **114** via the Internet **102**, for instance using GPRS, which the content server **114** then streams to the mobile terminal **103**. Reasonable quality video with mono audio can be obtained with a bit rate of 30 kbps. If the broadcast server **100** is arranged to receive broadcast television at the channel feeds **101**, the user can thus be provided with broadcast television on their mobile terminal **103**, and can change channel through a suitable provision on the user interface. This is achieved using GPRS as the bearer, and eliminates the need for the mobile terminal to be provided with broadcast television receiving hardware such as a DVB-T or DVB-H receiver. Similarly, very good quality stereo audio can be obtained with a bit rate of 30 kbps. 30 kbps has been found to be the maximum practical bandwidth with GPRS. 3G has been found to give practical bandwidths of 50-60 kbps. Multiple channels can be configured for each content source with different bitrates, one bit rate for GPRS, the other for 3G. This can also be made subscription tariff dependent.

[0167] This allows a user to receive broadcast radio services on the mobile terminal **103** through the broadcast server **100**, without the need for the mobile terminal **103** to be provided with FM or other radio receiver hardware. These bit rates are selected so as to provide a compromise between reliability of service, bearing in mind the 128 kbps maximum bit rate of GPRS and the likelihood of imperfect channel conditions, and quality of content reproduction. Clearly, better coding provides a better quality of content for a given bitrate, although the coding technique selected should not place an unnecessarily decoding burden on the mobile terminal **103**, since such is likely to increase the frequency of battery recharges.

[0168] The DRM encoder **113** adds digital rights management information to the content provided by the content encoder **112** such that only valid subscribers are able to properly decode the content streamed from the content server **114**. In particular, the DRM encoder **113** adds a digital fingerprint to the content stream at approximately regular intervals. The digital fingerprint can be removed only by valid subscribers. Failure to remove the digital fingerprint results in correct decoding of the content streams being impossible. Thus, the inclusion of the digital fingerprint prevents users other than valid subscribers watching the broadcast television channels and listening to the broadcast radio channels. Furthermore, a different digital fingerprint can be applied to different content streams, so restrictions which apply to some channels may not apply to other channels.

[0169] Details of a software media player **120** included within and installed on the mobile terminal **103** are shown in FIG. **12**. Referring to FIG. **12**, the media player **120** includes a connection to the Internet, through a GPRS connection through a mobile telephone network (not shown) associated with the mobile network operator with which the user of the mobile terminal **103** has a subscription or other contract. The media player **120** includes also a communications interface **121**, which feeds received streams to a DRM decoder **122**. A custom codec **123** and a Real™ or MP4 codec **124** are both fed by the DRM decoder **122**. Which of the Codecs **123** and **124** is used at a given time depends on the coding used at the broadcast server in respect of that channel. Both Real™ and MP4 codecs are able to process audio streams (for radio channels) as well as video streams (for television channels). Each of the codecs **123**, **124** has an output connected to a display engine and user interface **125**. A channel update module **126** is connected to the broadcast server **100** via the Internet **102**, and obtains information about the available channels therefrom. This channel information is stored in a channel store **127**. In response to a channel selection signal from the display engine and user interface **125**, the channel store **127** provides channel specification information to the display engine and user interface **125**. This channel specification information is passed from the display engine and user interface **125** to the communications interface **121**, which uses the channel specification information to ensure that it receives the correct content stream at any given time.

[0170] A billing verification module **128** is connectable to the billing module **106** and the WAP registration module **104** of the broadcast server **100** through the Internet **102**. These modules cooperate to register then subscribe the mobile terminal **103** to one or more services. The billing verification module uses the IMEI of the mobile terminal **103**, which is provided from an IMEI store **129** forming part of the mobile terminal, to identify the mobile terminal. Once a subscription has been set up, an access code is sent from the billing module **106** to the billing verification module **128**. This access code then is stored in a billing/DRM configuration store **130**. The access code includes the digital fingerprint and identifies the location of the fingerprint within the

content stream. The access code may relate to a single channel, or it may relate to a bundle of channels. The DRM decoder **122** is arranged to receive DRM information from the billing/DRM store **130**. Using this information, the DRM decoder **122** is able to remove the digital fingerprint from the content stream, which allows the content stream to be able to be decoded correctly by the custom codec **123** or the Real/MP4 codec **124**.

[**0171**] The communications interface **121** is arranged also to receive information from the billing/DRM configuration store **130**. This allows the media player **120** to register with the broadcast server **100** and to subscribe therewith. The media player **120** includes a menu option for registering for television and/or radio services on selection of this menu item, the media player starts a WAP session and connects to the registration module **104** of the broadcast server. Subscription and billing also is performed via WAP. Once registration and any subscription and/or billing is complete, the WAP session is ended and the media player **120** returns to allow its other functions to be selected. Billing is performed on a per channel per unit of time basis, and on a subscription basis. A subscription has a duration or an end date, and can relate to a single channel or to a package of channels.

[**0172**] Referring to FIG. **13**, a system **140** for providing a user with content comprises the broadcast server **100**, the Internet **102** and the mobile terminal of FIGS. **12** and **13**. The system also comprises a payment server **141**, which is connected to the Internet **102**. In this example, the payment server **141** is external to the broadcast server **100**. The payment server **141** may be operated by a different entity to the entity operating the broadcast server **100**. For instance, the payment server **141** may be operated by a mobile network operator. Also connected to the Internet **102** is a content server **142**, which may be operated by the operator of the broadcast server **100** or by a different operator. The content server **142** and the payment server **141** may be operated by the same operator. The GPRS network which connects the mobile terminal **103** to the Internet **102** is shown at **143** in the Figure.

[**0173**] A detailed description of the use of digital fingerprints and obfuscation in the FIGS. **12** and **13** system will now be described. When streaming content, via GPRS or any other carrier, the data content blocks are transmitted over a different channel to the corresponding headers. Any suitable channels are used for this purpose.

[**0174**] The headers and the content blocks have the same content as those shown in and as described above with reference to FIG. **9**, although the headers and content blocks are not sequential as shown. The headers each have a one-to-one relationship with content blocks, and are in the same sequence. No index headers or index entries are present. Since the transmission can be a continuous process, there are no physical start addresses associated with the content blocks, not are there first headers or content blocks.

[**0175**] Each of the content blocks **1204-1208** relates either to audio or video content. The video content blocks form a different stream to the audio content blocks. Each content block includes a presentation timestamp, so that a media player can relate audio content to the corresponding video content. The header of a content block **1204-1208** includes information which identifies which stream the content block relates to. In most cases, there will be only one audio stream and only one video stream, although this may vary.

[**0176**] In this example, the content blocks are produced by a RealProducer tool, so the headers **1201-1203** and the data in the content blocks **1204-1208** are compliant with the Real format. The content is decodable by a Real codec. Real is a trademark of RealNetworks, Inc. With Real, each content block **1204-1208** which relates to a video stream contains data relating to a whole video frame. Thus, each content block includes all the information relating to a frame, and relates to only one frame. A key frame is provided periodically, and each non-key frame between successive key frames detail differences between the previous frame and that frame. The interval between successive key frames is a settable parameter in RealProducer. The size of content blocks is variable. Where there is a lot of difference between two successive frames, the content block for the second frame includes a relatively large amount of data. Where there is less difference, the second frame typically contains less data. Where there is no difference between successive frames, the second content block may contain only a few bytes of data. Each header **1201-1203** includes information which identifies whether or not the corresponding content block **1204-1208** includes data of a key frame.

[**0177**] The offset to the start of content blocks is defined in a media header, which is sent first. Content block headers are constant length and define the variable length of the content block data. Content blocks are contiguous. Since the media player knows the offset to the first block from the main header and then the offset to the following blocks, the media player is able to determine the start of content blocks in the stream. The end of content is identified when the stream terminates or when an identifier for the index section is read as the start of the next content block header.

[**0178**] It is not possible for a machine or human operator to determine from examining the data of the content blocks **1204-1208** where the boundaries between the content blocks lie. However, each of the content headers **1201-1203** includes data which indicates the length in bytes of the corresponding content block **1204-1208**. Thus, the start address of a content block **1204-1208** can be determined by adding the start address of the previous content block the length of the previous content block and one additional byte. Adding the length of the previous block to the start address of that block results in the address of the final byte of that block, thus the start address of the following content block is found by adding an additional byte to that address.

[**0179**] Thus, if any bits or bytes are added or taken away from any of the content blocks **1204-1208**, the actual start addresses of all following content blocks will not match the addresses calculable from the headers **1201-1203**. In this event, the data fed into a Real decoder will not be decoded properly, and the media player would probably stop working altogether and require closing and reopening before it could become operational.

[**0180**] The inventors exploit this to advantage. In particular, some content blocks **1204-1208** are each provided with one or more excess data items. The excess data item preferably is a digital fingerprint. In this example, the 2nd, Xth and Yth content blocks **1205**, **1206**, **1207** are provided with a digital fingerprint.

[**0181**] It clearly is important for a media player which is to playback the content to know where the excess data items

are. Otherwise, data sent to a decoder in the media player may not be in the correct format, and thus will not be decoded properly and may crash the media player.

[0182] In some embodiments, excess data items are included at regular intervals (in a playback or presentation time sense) in the stream. For instance, excess data items can be included in the first content block having a timestamp immediately following an integer multiple of a predetermined excess data item interval, for instance 20 seconds. According to this technique, excess data items are included in the first content block having a presentation time after 20 seconds, in the first content block having a presentation time after 40 seconds, the first content block having a presentation time after 60 seconds and so on. Although the stream can be continuous, the presentation timestamps eventually cycle around to zero. However, it is a straightforward issue to determine whether a given timestamp is the first timestamp after an integer multiple of an excess data item interval.

[0183] The location of the excess data within the content block **1205-1207** also is important. In some embodiments, the excess data items are included at the same position in each of the content blocks **1205-1207** in which it present. For instance, the excess data may be included from byte **56** of those content blocks **1205-1207**. Bytes **57** onwards of the content block are retained, but after the digital fingerprint. If a content block includes 56 content blocks or fewer, then the excess data items are included at the end of that content block, after all of the content data. Thus, if a content block includes 20 bytes, then the excess data begins at the 21st byte.

[0184] There are numerous ways in which a media player can be informed of the locations of excess data items in content data. The inventors have found that a particularly good solution is to utilise a pre-determined scheme to determine where to include excess data items before streaming the data, and to program the media player with details of that scheme.

[0185] For instance, there may be twenty different schemes that the media player is programmed to handle. Each scheme is identified by a different digital fingerprint scheme identifier. This may form part of the media code or may be identified to the media player in any other suitable way. A first scheme may involve digital fingerprints included at byte **56** in each of the first content blocks following integer multiples of 20 seconds, as described above. A second scheme may involve including a digital fingerprint at byte **32** of every content block immediately following integer multiples of 30 seconds. A third scheme may involve including a digital fingerprint at byte **32** of the content block six content blocks following integer multiples of 30 seconds. A fourth scheme may involve including a digital fingerprint at byte **32** of every content block immediately following integer multiples of 60 seconds and at byte **11** of every content block immediately following integer multiples of 30 seconds unless there is a digital fingerprint at byte **32** thereof, i.e. the location of the digital fingerprint alternates between byte **11** and byte **32** between successive occurrences of it. In another scheme, the content block in which the fingerprint is located is varied. For instance, the digital fingerprint may be located relative to a content block immediately following integer multiples of 40 seconds, at three content blocks following, seven content blocks following and two content blocks following in a repeated sequence. The more the location of the digital fingerprint is varied, the greater the protection that is afforded.

[0186] The length of the excess data items are not critical, although to avoid increasing the size of the resulting data by a significant degree the excess data preferably is not unduly long.

[0187] Since the excess data items are intended to be removed before decoding, the form (i.e. content) of the excess data items are not necessarily important. However, the inventors prefer that the excess data items are in the form of a digital fingerprint. Preferably, each occurrence of the digital fingerprint is the same, i.e. has the same data sequence. For instance, the digital fingerprint may be 5 bytes long. Even if a third party manages to determine the data constituting the digital fingerprint, data strings having the same data will be present at numerous locations in the content data, so this information alone would not be enough to allow the digital fingerprints to be removed.

[0188] As shown in FIG. **10**A, an unmodified content block **1206** includes m bytes of data. In FIG. **10**B, the content block **1206** is shown with a digital fingerprint **1301** added, and is labelled **1304**. The digital fingerprint **1301** separates the m data bytes into two sections **1302**, **1303**. The first section **1302** includes data bytes **0** to n, and the second section **1303** includes data bytes n+1 to m. The length of the content block with the fingerprint is equal to m plus k bytes, where k is the size of the digital fingerprint. Using the example given above, n is 56 and k is 5.

[0189] To provide additional protection against unauthorised playback and/or copying, some or all of the content blocks are obfuscated before they are streamed. Where a content block includes an excess data item, such as a digital fingerprint **1301**, then the excess data item is obfuscated along with the data forming the original content block. Simply, obfuscation comprises altering the data so that the resulting obfuscated data is different to the original data and cannot be decoded properly without first being deobfuscated. Obfuscation typically does not alter the amount of data, so the size of a content block is the same before and after obfuscation. Obfuscation is discussed above in relation to FIG. **3**. The content block **1304** including the digital fingerprint **1301** is shown obfuscated at **1305** in FIG. **10**C. The obfuscated content block **1305** includes k plus m bytes, as with the digitally fingerprinted content block **1304**.

[0190] If only one obfuscation scheme is used with all content, then the media player **58** knows what obfuscation is used without being informed of this.

[0191] If the obfuscation scheme is not the same for all content, then the media player **58** may need to be informed which obfuscation scheme is used with the content. In this case, an obfuscation identifier can be obtained from the broadcast server **100** in any suitable way. The obfuscation identifier **1306** identifies what obfuscation method was used by the DRM encoder **113** to obfuscate the data. This allows the media player it to perform the corresponding de-obfuscation method.

[0192] The content blocks **1203-1208** which include a digital fingerprint include more data than is indicated by their corresponding header **1201-1203**

[0193] Following obfuscation of some or all of the content blocks **1205-1208**, the resulting data is streamed to the mobile device **103**.

[0194] When a user operates the media player on their mobile device **103** to receive a streamed television or radio channel, the mobile device communicates with the broadcast server **100** and arranges for an appropriate stream to be delivered to the mobile device. The headers and any obfuscation identifier are received separately. The media player then begins to read the headers and the content blocks relating to that content. The DRM decoding module **122** within the media player uses any obfuscation identifier to determine what obfuscation method is needed to de-obfuscate the content data **57**. The DRM decoding module **122** also determines whether the media code needed to render the streamed content is present, having been obtained already during an authorisation process. The media code identifies the digital fingerprint **1301**, and allows the media player **58** to determine in which content blocks **1205-1208** and where in those content blocks the digital fingerprint is present. This can occur in any suitable way. For instance, the media code may include a digital fingerprint location code, which identifies a predetermined scheme useable to remove the occurrences of the digital fingerprint.

[0195] The media player then is ready to render the streamed content. However, the media player is arranged to playback the content only if the digital fingerprint included in the streamed content blocks is the same as the digital fingerprint included in the media code. This provides a further check that the user is entitled to playback the content using the media player.

[0196] Without knowing where fingerprint is, it cannot be removed from the content by the media player. Also, the media player ensures that fingerprint present in the content is as expected before it will play the content.

[0197] In playing back the content, the DRM decoding module **122** de-obfuscates those content blocks which are obfuscated. For instance, in respect of the content block **1305** of FIG. **10**C, the DRM validation module **44** performs the inverse of the obfuscation performed at the DRM processing module **21**, thereby obtaining the fingerprinted content block **1304**. In playing back the content, the DRM processing module **21** also removes fingerprints from the content blocks that include them. For instance, in respect of the content block **1304** of FIG. **10**B, the DRM processing module **21** removes the digital fingerprint **1301**, thereby obtaining the content block **1206** shown at FIG. **10**A. The content blocks are fed to the codec **124** or the codec **123** of the media player only after de-obfuscation and after removal of the digital fingerprints. Failure to do either of these actions would result in incorrect data being fed to the codec, likely resulting in the crashing of the media player. At best, content would not be played back in a useable form.

[0198] If any other media player is used to attempt to playback the content, it will fail. In order to construct a media extractor which could extract useable content, the media extractor would need to know exactly what obfuscation method to use, and exactly where in the content blocks the excess data items are and what size they are. Thus, this technique provides substantial protection against unauthorised use of the content. Even with a relatively simple obfuscation method and relatively infrequent digital fingerprint inclusion, the protection afforded is relatively strong. Since the media player knows what de-obfuscation method is used and from what locations the digital fingerprint needs to be removed, it can playback the content correctly. However, the media player is not a media extractor, so cannot be used to extract the content in unprotected form for unauthorised use.

Obtaining Content

[0199] It has been known for some time to stream video or audio to a personal computer connected to the Internet, with the computer being used to reproduce the content for viewing/listening by a user. It is known now to stream audio and video clips to a mobile terminal on-demand, and to use a software media player on the terminal to reproduce the content. This functionality is known from, e.g. the MM-A**700** mobile phone produced by Samsung. Streamed video content is provided by Sprint PCS Vision Multimedia Services. It is known also to provide mobile telephones and other hand-portable devices with MP3 and similar players, which produce music and other audio content from compressed data pre-loaded onto the terminal or onto a removable memory device connected with the terminal.

[0200] Streaming content to mobile terminals allows greater opportunity for users to be exposed to music and other content which is new to them and in which they may be interested.

[0201] According to another aspect of the present invention, there is provided a method of providing an item of content to a terminal, the method comprising:

[0202] streaming content to a terminal,

[0203] at the terminal, detecting a user input indicating that the user wants a content item which is at least one of:

[0204] a) forming part of the streamed content, and

[0205] b) related to content forming part of the streamed content,

[0206] in response to the user input, sending a request from the terminal to a server;

[0207] in response to receiving the request,

[0208] identifying the content item that is required,

[0209] determining whether the user is entitled to that content item, and

[0210] providing content obtaining means to the terminal; and

[0211] in response to receiving the content obtaining means at the terminal, controlling the terminal to use the content obtaining means to obtain the content item.

[0212] According to another aspect of the invention, there is provided a system comprising a server, a terminal and means for streaming content to the terminal,

[0213] the terminal being operable in response to a user input indicating that the user wants a content item which is at least one of:

[0214] 5 a) forming part of the streamed content, and

[0215] b) related to content forming part of the streamed content,

[0216] to send a request to the server;

[0217] the server being responsive to receiving the request to identify the content item that is required, to determine whether the user is entitled to that content item, and to provide to the terminal content obtaining means; and

[0218] the terminal being operable to use the content obtaining means to obtain the content item.

[0219] According to another aspect of the invention, there is provided a terminal comprising:

[0220] means for receiving streamed content,

[0221] means responsive to a user input indicating that the user wants a content item which is at least one of: a) forming part of the streamed content, and b) related to content forming part of the streamed content, to send a request to a server; and

[0222] means responsive to receiving content obtaining means to use the content obtaining means to obtain the content item.

[0223] According to another aspect of the invention, there is provided a method of operating a terminal, the method comprising:

[0224] receiving streamed content,

[0225] in response to a user input indicating that the user wants a content item which is at least one of: a) forming part of the streamed content, and b) related to content forming part of the streamed content, sending a request to a server; and

[0226] in response to receiving content obtaining means, using the content obtaining means to obtain the content item.

[0227] These aspects of the invention allow users to obtain content items at their terminal triggered whilst receiving streamed content at the terminal. This can make the process of obtaining, e.g. by purchase, content that the user is interested in straightforward for the user, especially since there is no need to determine what content is being streamed and then independently obtain that content from another source. The process can be technically straightforward as well, allowing system components which generally are unrelated to each other and which are not designed for interoperability to cooperate to allow a user to be provided with content.

[0228] Embodiments of these other aspects invention will now be described, by way of example only, with reference to the accompanying drawings, of which:

[0229] FIG. **11** is a schematic diagram illustrating components of a broadcast server operable with a media player according to the present invention; and

[0230] FIG. **12** is a schematic block diagram illustrating components of a media player operable according to certain aspects of the present invention.

[0231] FIG. **13** is a schematic diagram of components of a system through which a terminal is able to obtain content according to the invention and including components according to the invention; and

[0232] FIG. **14** is a flow chart illustrating operation of the FIG. **3** system when operating to provide content to a terminal, according to various aspects of the invention.

[0233] An overview of the broadcast television and radio system will now be described with reference to FIGS. **11** and **12**. Referring to FIG. **11**, a broadcast server **100** includes an input **101** at which channel feeds are received, and input/outputs to the Internet **102**. A mobile terminal **103** is connected to the Internet **102** through a mobile network (not shown), and thus is able to communicate with the broadcast server **100**. Briefly, once a user of the mobile terminal **103** has subscribed to broadcast services, they are able to request streaming to them of data through which they can view a television channel or listen to a radio channel using the mobile terminal **103**.

[0234] The broadcast server **100** includes a WAP registration module **104**, through which a user of the mobile terminal **103** can become registered with the broadcast server **100** through a WAP connection to the Internet **102**. The mobile terminal **103** may be identified for example by its IMEI number. The registration module is in two-way communication with a registration database **105**, which maintains details of registered users and which allows a supervisor to monitor registered users and to unregister them as required. Following registration, the user is able to subscribe to services using a WAP connection between a billing module **106** and the Internet **102**. The billing module **106** is in two-way communication with a billing database **107**, which monitors subscriptions and allows a supervisor to examine individual subscriptions and to provide subscription statistics. The billing database **107** and the registration database **105** are in two-way communication with one another, so that registration information can be passed to the billing module **106** and subscription and billing information can be passed to the registration database **105** and the registration module **104**.

[0235] A channel configuration database **108** maintains configuration parameters of channels between the broadcast server **100** and multiple mobile terminals, only one of which is shown at **103** in the Figure. Channel configurations are passed from the configuration database to a channel configurator **109**, which has an http connection to the Internet **102** The channel configuration database **108** contains configuration data for the channels. The channel configuration database **108** is updated using a web based administration tool to add, modify and remove channels to conform to incoming streams, which are setup by a manual configuration process.

[0236] The data included in the channel configuration database **108** consists of the full URL of each channel.

[0237] The channel configurator **109** reads the channel configuration database **108**, prepares an XML list of all channels available to a particular user (i.e. the channels to which they have subscribed) and sends this XML list to the media player on the mobile device. A menu option to "refresh channels" within the media player can be used to initiate this process. The media player then creates a new channel list for the user.

[0238] Data received at the channel feeds input is processed by a chain of components comprising stream converters **110**, stream buffers **111**, a content encoder **112**, a DRM encoder **113** and a content server **114**. In this example, the streamed content is in Real™ format, so the content encoder **112** is RealProducer™ and the content server **114** is RealServer™, although any other suitable format may be used instead. There is a stream converter **110** and a steam buffer **111** for each incoming channel feed at the input **101**. The stream converters **110**, the content encoder **112** and the DRM encoder **113** receive channel configuration information from the channel configuration database. The DRM encoder **113** also receives subscription information from the billing database **107**.

[0239] The content server **114** supplies streamed channels to the Internet **102**, from where they can be accessed by the mobile terminal **103** by applying a stream request to the content server **114** via the Internet **102**.

[0240] Using the broadcast server **100**, the user of the mobile terminal **103** is able to register and subscribe to services. When subscribed, the user of the mobile terminal **104** is able to select a channel from the content server **114** via the Internet **102**, for instance using GPRS, which the content server **114** then streams to the mobile terminal **103**. Reasonable quality video with mono audio can be obtained with a bit rate of 30 kbps. If the broadcast server **100** is arranged to receive broadcast television at the channel feeds **101**, the user can thus be provided with broadcast television on their mobile terminal **103**, and can change channel through a suitable provision on the user interface. This is achieved using GPRS as the bearer, and eliminates the need for the mobile terminal to be provided with broadcast television receiving hardware such as a DVB-T or DVB-H receiver. Similarly, very good quality stereo audio can be obtained with a bit rate of 30 kbps. 30 kbps has been found to be the maximum practical bandwidth with GPRS. 3G has been found to give practical bandwidths of 50-60 kbps. Multiple channels can be configured for each content source with different bitrates, one bit rate for GPRS, the other for 3G. This can also be made subscription tariff dependant.

[0241] This allows a user to receive broadcast radio services on the mobile terminal **103** through the broadcast server **100**, without the need for the mobile terminal **103** to be provided with FM or other radio receiver hardware. These bit rates are selected so as to provide a compromise between reliability of service, bearing in mind the 128 kbps maximum bit rate of GPRS and the likelihood of imperfect channel conditions, and quality of content reproduction. Clearly, better coding provides a better quality of content for a given bitrate, although the coding technique selected should not place an unnecessarily decoding burden on the mobile terminal **103**, since such is likely to increase battery recharge intervals.

[0242] The DRM encoder **113** adds digital rights management information to the content provided by the content encoder **112** such that only valid subscribers are able to properly decode the content streamed from the content server **114**. In particular, the DRM encoder **113** adds a digital fingerprint to the content stream at approximately regular intervals. The digital fingerprint can be removed only by valid subscribers. Failure to remove the digital fingerprint results in correct decoding of the content streams being impossible. Thus, the inclusion of the digital fingerprint prevents users other than valid subscribers watching the broadcast television channels and listening to the broadcast radio channels. Furthermore, a different digital fingerprint can be applied to different content streams, so restrictions which apply to some channels may not apply to other channels.

[0243] Details of a software media player **120** included within and installed on the mobile terminal **103** are shown in FIG. **12**. Referring to FIG. **12**, the media player **120** includes a connection to the Internet, through a GPRS connection through a mobile telephone network (not shown) associated with the mobile network operator with which the user of the mobile terminal **103** has a subscription or other contract. The media player **120** includes also a communications interface **121**, which feeds received streams to a DRM decoder **122**. A custom codec **123** and a Real™ or MP4 codec **124** are both fed by the DRM decoder **122**. Which of the Codecs **123** and **124** is used at a given time depends on the coding used at the broadcast server in respect of that channel. Both Real™ and MP4 codecs are able to process audio streams (for radio channels) as well as video streams (for television channels). Each of the codecs **123**, **124** has an output connected to a display engine and user interface **125**. A channel update module **126** is connected to the broadcast server **100** via the Internet **102**, and obtains information about the available channels therefrom. This channel information is stored in a channel store **127**. In response to a channel selection signal from the display engine and user interface **125**, the channel store **127** provides channel specification information to the display engine and user interface **125**. This channel specification information is passed from the display engine and user interface **125** to the communications interface **121**, which uses the channel specification information to ensure that it receives the correct content stream at any given time.

[0244] A billing verification module **128** is connectable to the billing module **106** and the WAP registration module **104** of the broadcast server **100** through the Internet **102**. These modules cooperate to register then subscribe the mobile terminal **103** to one or more services. The billing verification module uses the IMEI of the mobile terminal **103**, which is provided from an IMEI store **129** forming part of the mobile terminal, to identify the mobile terminal. Once a subscription has been set up, an access code is sent from the billing module **106** to the billing verification module **128**. This access code then is stored in a billing/DRM configuration store **130**. The access code includes the digital fingerprint and identifies the location of the fingerprint within the content stream. The access code may relate to a single channel, or it may relate to a bundle of channels. The DRM decoder **122** is arranged to receive DRM information from the billing/DRM store **130**. Using this information, the DRM decoder **122** is able to remove the digital fingerprint from the content stream, which allows the content stream to be able to be decoded correctly by the custom codec **123** or the Real/MP4 codec **124**.

[0245] The communications interface **121** is arranged also to receive information from the billing/DRM configuration store **130**. This allows the media player **120** to register with the broadcast server **100** and to subscribe therewith. The media player **120** includes a menu option for registering for television and/or radio services on selection of this menu

item, the media player starts a WAP session and connects to the registration module **104** of the broadcast server. Subscription and billing also is performed via WAP. Once registration and any subscription and/or billing is complete, the WAP session is ended and the media player **120** returns to allow its other functions to be selected. Billing is performed on a per channel per unit of time basis, and on a subscription basis. A subscription has a duration or an end date, and can relate to a single channel or to a package of channels.

[**0246**] Referring to FIG. **13**, a system **140** for providing a user with content comprises the broadcast server **100**, the Internet **102** and the mobile terminal of FIGS. **11** and **12**. The system also comprises a payment server **141**, which is connected to the Internet **102**. In this example, the payment server **141** is external to the broadcast server **100**. The payment server **141** may be operated by a different entity to the entity operating the broadcast server **100**. For instance, the payment server **141** may be operated by a mobile network operator. Also connected to the Internet **102** is a content server **142**, which may be operated by the operator of the broadcast server **100** or by a different operator. The content server **142** and the payment server **141** may be operated by the same operator. The GPRS network which connects the mobile terminal **103** to the Internet **102** is shown at **143** in the Figure.

[**0247**] Operation of the components of the system **140** will now be described with reference to FIG. **13**. Operation begins at step S**1** with the mobile terminal **103** receiving an audio stream from the broadcast server **100**. Here, the user can be listening to a radio station whose content is streamed over the Internet **102** and through GPRS to the mobile terminal **103**. The radio channel typically includes a number of music tracks, or 'songs' played sequentially, occasionally interspersed with talk, 'jingles' and/or advertisements. At step S**2**, the user indicates that he or she wants to purchase the track that is currently being played by the radio station. This can occur in any convenient manner. For maximum convenience to the user, the media player **120** residing on the mobile terminal **103** is operable when playing a radio station to provide a soft key or other convenient key as a 'purchase track' option selector. Thus, whenever a user is listening to a radio station through the mobile terminal **103**, the user is able to indicate that a track is required to be purchased in a quick and straightforward manner. To reduce the possibility of a user accidentally selecting a track for purchase, the media player **120** is arranged to require the user to confirm that a track is to be purchased before proceeding with purchase, for example by pressing a different key designated by the media player **120** as a confirmation key. The requirement for pressing of the confirmation key, as well as the identity of the confirmation key, is indicated on the display of the mobile terminal **103**.

[**0248**] The requesting of the track by the user at step S**2** causes the media player **120** to send an http request at step S**3** through the GPRS network **143** and the Internet **102** to the broadcast server. The http request of step S**3** optionally includes a timestamp or other data which indicates the absolute time at the time of the user requesting the track at step S**2**, or else data from the streamed content which could be used by the broadcast server to identify the time at which the user requested the track at step S**2**. If however it can be assumed that there is no significant delay between step S**2** and the broadcast receiving the http request of step S**3**, then no timestamp or other time data needs to be included.

[**0249**] On receipt of the http request, the broadcast server **100**, in particular the content server **114** thereof, identifies at step S**4** the track that is currently being played on the radio channel that is being streamed to the mobile terminal. If the http request includes a timestamp or other time data, then step S**4** comprises determining the identity of the track that was playing at the time that the user requested the track at step S**2**. Where the broadcast server **100** produces the radio station itself, then the track identity is already available to it. Where the broadcast server **100** is streaming a radio station which is being received from an external source, then this step may involve for example accessing a website operated by the radio station to determine the identity of the current track.

[**0250**] Following step S**4**, the broadcast server **100** identifies a product code for the requested track at step S**5**. This can occur in any suitable way, for example using a look-up table. The product code uniquely identifies the requested track. It may take any suitable for, and may originate from any suitable source. The product code may originate from the content server **142**. Following step S**5**, the broadcast server creates a payment URL (uniform resource locator) at step S**6**. The payment URL identifies a resource that the mobile terminal **103** can visit in order to obtain clearance to obtain the content. The payment URL and the product code are then sent to the mobile terminal at step S**7** as a response to the http request sent at step S**3**. They are received by the mobile terminal **103** at step S**8**.

[**0251**] At step S**9**, the mobile terminal **103** attempts to access the resource at the URL received at step S**8**. The URL relates to a web or WAP page stored at the payment server **141**. The attempt to access the resource involves the mobile terminal **103** sending an http request at step S**10** over the Internet **102** to the payment server **141**. In response, the payment server **141** prepares a web or WAP page at step S**11**, and sends the corresponding data at step S**12** to the mobile terminal **103**. The web or WAP page includes a data entry field, into which the mobile terminal **103** under control of the media player **120** automatically inserts the product code at step S**13** and sends the appropriate data to the payment server **141** at step S**14**. The product code thus is received by the payment server **141** at step S**15**. The payment server **141** knows from the data received at steps S**10** and S**14** what is the identity of the mobile terminal **103**.

[**0252**] The payment server **141** then processes the payment at step S**16**. This may take any suitable form. For instance, the payment server **141** may relate to a service with which the user of the mobile terminal **103** has registered a credit or other payment card. In this case, processing the payment may involve the payment server **141** merely debiting the payment card by a suitable amount. Alternatively, the user may have prepaid an amount, in which case the payment server **141** may merely deduct a suitable amount from the user's prepaid account. Alternatively, the payment server **141** may be operated by the user's mobile network operator, in which case a suitable amount may be added to the user's mobile telephone bill.

[**0253**] It will be appreciated that payment is not necessary if, for example, the owner of the copyright in the requested track does not require payment for a user's licence, or if the

user has an allowance of, for example, three free tracks per month and the user has one or more free tracks remaining for the current month.

[0254] Whether or not payment by the user is required, the result of the step of processing payment at step S15 is either that payment failed, in which case the user is not granted access to the requested track, or that payment succeeds, in which case the user of the mobile terminal 103 is deemed to be entitled to the track. Step S17 relates to the situation in which the user is entitled to the track.

[0255] The payment complete step S17 triggers two actions. Firstly, a token is sent at step S18 to the broadcast server 100. The broadcast server 100 then confirms that the token indicates that the user of the mobile terminal 103 is entitled to the track. The token includes an identifier of the mobile terminal 103 and/or a transaction identifier so that the broadcast server 100 can distinguish between different track requests. In response to this confirmation, the broadcast server 100 sends confirmation to the payment server 141 at step S20. In response to receiving the confirmation from the payment server 141, the broadcast server 100 ends its involvement in the process. The other action triggered by the payment complete step S17 is the sending of a content obtaining URL from the payment server 141 to the mobile terminal 103, indicated at step S21 in the Figure.

[0256] The content obtaining URL includes a link to a web or WAP page maintained by the broadcast server 100. Following receipt of it at the mobile terminal 103, the media player 120 causes the browser of the mobile terminal 103 to be directed to the resource addresses by the URL. This is indicated at step S23 in the Figure. In the meantime, following the confirmation step S19, the broadcast server 100 at step S24 prepares XML data which when provided to the mobile terminal 103 allows it to obtain the content from the content server 142. In response to the mobile terminal 103 accessing the page at the content obtaining URL, the broadcast server 100 sends the XML data to the mobile terminal 103 at step S25.

[0257] The mobile terminal 103 is controlled by the media player 120 automatically on receiving XML data from the broadcast server 100 to follow the instructions therein. This includes controlling the mobile terminal 103 at step S26 to contact the content server 142 in such a way and with such data that the content server can identify the content, for example from the product code sent from the broadcast server at step S7 and included within the XML data prepared at step S24, and verify that the mobile terminal 103 is entitled to the requested content. Thus, the step S26 of the mobile terminal 103 retrieving the content is cooperative with a step S27 of the content server 142 providing the content. Following the provision of the content by the content server 142 to the mobile terminal 103, both of those components end their involvement in the process.

[0258] Some mobile terminals may not be capable of using XML data to retrieve content from the content server 142. In this case, a different technique needs to be used instead.

[0259] In addition to allowing the user to request the current track at step S2, the media player 120 may allow the user instead to request the previous track, or another identifiable track. This involves the provision of a separate input on the user interface provided by the media player 120, which is a subsidiary option to the option of requesting the current track. In this case, the broadcast server 100 is provided with suitable functionality.

Providing Audio-Visual Content

[0260] Further aspects of the invention relate to apparatus for providing audio-visual content for reproduction on a mobile device, to a method of providing audio-visual content for reproduction on a mobile device, to data stored on a portable medium or existing at least transiently in memory, and to a method of operating a mobile device.

[0261] There is a trend for mobile telephones, also known as cellular telephones, to be provided with colour displays having many thousands of pixels. As time progresses the quality of these displays and the resolutions afforded thereby increases. Furthermore, semiconductor terminology is such the mobile telephones can be provided with quite substantial amounts of memory. Whereas previously it has been known to incorporate MP3 players and the like into mobile telephones, the provision of improved displays and increased amounts of memory allows mobile telephones to be used for use as limited digital television receivers. It has been proposed as well to provide audio-visual content on a multimedia card (MMC), for viewing on a mobile telephone. The Nokia™ 7610 is one such capable mobile telephone. This telephone can handle 3GPP and RealMedia audio-visual formats.

[0262] Providing audio-visual content for consumption on a mobile device currently is a laborious and time-consuming process. It is an aim of the present invention to provide apparatus and a method for providing audio-visual content for reproduction on a mobile device which is convenient yet capable of utilising the full capabilities of a target mobile device.

[0263] According to a further aspect of the invention, there is provided apparatus for providing audio-visual content for reproduction on a mobile device, the apparatus comprising:

[0264] an audio-visual content supply arrangement;

[0265] the apparatus being arranged to write into an area of memory data constituting:

[0266] audio-visual content;

[0267] two or more different media players; and

[0268] a loader program,

the loader program being arranged such that when loaded into a mobile device it causes configuration parameters of the mobile device to be determined, causes one of the media players to be selected on the basis of the detected configuration parameters, and controls the mobile device to use the selected media player.

[0269] In this way, it is possible to utilise for example an MMC card for a greater number of target device configurations. This can be advantageous, especially when for instance MMC cards are intended for retail from a shop display or similar.

[0270] The loader program may be arranged to control the mobile device to detect whether or not it already includes a

suitable media player and, if a suitable media player is detected, this is controlled to be used instead of installing a media player from the area of memory onto the mobile device.

[0271] This can be advantageous since it can reduce the possibility of there being an installation or deinstallation error, thereby improving the reliability of the mobile device.

[0272] The two or more media players may be provided through a single configurable media player software application. In this case, the loader program is operable to determine what media player is required, and to operate appropriate software modules forming part of the media player software. Thus, multiple media players can be made up from a single software application, which reuses modules or functions for certain media player functionality.

[0273] If the two or more media players are provided through a single configurable media player software application, the loader program may form part of the media player software application.

[0274] According to a further aspect of the invention, there is provided data stored on a portable medium or existing at least transiently in memory, the data constituting:

[0275] audio-visual content;

[0276] two or more different media players; and

[0277] a loader program,

the loader program being arranged such that when loaded into a mobile device it causes configuration parameters of the mobile device to be determined, causes one of the media players to be selected on the basis of the detected configuration parameters, and controls the mobile device to use the selected media player.

[0278] According to a further aspect of the invention, there is provided a method of providing audio-visual content for reproduction on a mobile device, the method comprising:

[0279] writing into an area of memory data constituting:

[0280] audio-visual content;

[0281] two or more different media players; and

[0282] a loader program,

the loader program being arranged to cause a mobile device to determine configuration parameters of the mobile device, to select one of the media players on the basis of the detected configuration parameters, and to control the mobile device to use the selected media player.

[0283] According to a further aspect of the invention, there is provided a method of operating a mobile device, the method comprising:

[0284] storing audio-visual content data and two or more different media players in internal and/or external memory;

[0285] determining configuration parameters of the mobile device,

[0286] selecting one of the media players on the basis of the detected configuration parameters, and

[0287] using the selected media player to consume the audio-visual content data.

[0288] The term 'mobile device' will be understood to embrace mobile (cellular) telephones and personal digital assistants having bidirectional voice communication capabilities, as well as other mobile devices, including dedicate media players and the like.

[0289] Embodiments of the further aspects of the present invention will now be described by way of example only, with reference to the accompanying drawings in which:

[0290] FIG. **1** is a schematic diagram of audio-visual content provision apparatus embodying the invention;

[0291] FIGS. **2** and **3** are flowcharts illustrating steps of operation of the FIG. **1** apparatus;

[0292] FIG. **4** is a schematic drawing illustrating apparatus for playback of the converted audio-visual content in a mobile telephone; and

[0293] FIG. **15** is a schematic drawing of a system of interconnected computers operable according to the invention.

[0294] Referring firstly to FIG. **1**, content extracting and converting apparatus **10** is illustrated schematically. Two alternative sources of audio-visual content **8**, **9** are included. A first content source **8** utilises film or movie data stored on a DVD (digital video disk or digital versatile disk) **15**. An automated extraction configuration module **16** examines metadata stored on the DVD **15** to determine the configuration of content data stored on the DVD. This involves the application of a tcprobe, and an analysis of the information returned from the DVD **15**. This is described in more detail below. The result is data stored in an extraction configuration memory area **17** representing an extraction configuration. The extraction configuration data from the memory area **17** is utilised by a DVD decryption and extraction module **18** to extract movie data (i.e. the content data) from the DVD **15**. The result is content data in an intermediate format, which is written to an intermediate format movie data area **14**. The data included in the intermediate format movie data area **14** is in predetermined format and is suitable for conversion into a form ready for reproduction on a mobile telephone (not shown). Preferably the intermediate format is AVI. This format has the advantage of high resolution, yet is relatively easy to handle and it is relatively easy to convert from AVI into 3GPP and many other formats suitable for use by mobile devices.

[0295] The second source of audio-visual content **9** receives from a movie data storage area **12** data representing a movie (or film) in AVI (audio-visual interleave) or other format. The movie so supplied is converted by a format conversion module **13** before being written to the intermediate format movie data area **14**.

[0296] Thus, either of the audio-visual content sources **8**, **9** can be used to provide movie data in the intermediate format movie data area **14**.

[0297] A mobile format conversion module **19** converts movie data stored in the extracted movie data area **14** and provides a movie in mobile telephone consumable format in a mobile format movie data area **20**. The mobile format conversion module **19** utilises a digital rights management

(DRM) processing module **21**, which allows certain control over the access and distribution of the resulting movie data. The conversion effected by the mobile format conversion module **19** uses a codec **22**, which preferably is custom-designed for the purpose. Importantly, the conversion effected by the mobile format conversion module **19** uses information stored in a production configuration data area **23**. By controlling the mobile format conversion module **19** on the basis of information specific to the configuration of, and thus tailored to, a target device, the apparatus **10** can be used to provide movie data for any of potentially a large number of target mobile devices.

[**0298**] The extraction effected by the audio-visual content source **12** will now be described in detail with reference to FIG. **2**.

[**0299**] In FIG. **2**, extraction configuration is effected at step S**1**. This utilises the automated extraction configuration **16** shown in FIG. **1**. Extraction configuration commences by analysing the DVD source **15**. The result of an example analysis, i.e. what is returned in response to a query, is illustrated below:

(dvd_reader.c) mpeg2 pal 16:9 only letterboxed U0 720x576 video

(dvd_reader.c) ac3 en drc 48 kHz 6Ch

(dvd_reader.c) ac3 de drc 48 kHz 6Ch

(dvd_reader.c) ac3 en drc 48 kHz 2Ch

(dvd_reader.c) subtitle 00=<en>

(dvd_reader.c) subtitle 01=<de>

(dvd_reader.c) subtitle 02=<sv>

(dvd_reader.c) subtitle 03=<no>

(dvd_reader.c) subtitle 04=<da>

(dvd_reader.c) subtitle 05=<fi>

(dvd_reader.c) subtitle 06=<is >

(dvd_reader.c) subtitle 07=<en>

(dvd_reader.c) subtitle 08=<de>

[tcprobe] summary for /media/dvdrecorder/, (*)=not default, 0=not detected

import frame size: -g 720x576 [720x576]

[**0300**] aspect ratio: 16:9 (*)

[**0301**] frame rate: -f 25.000 [25.000] frc=3

[**0302**] audio track: -a 0 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[**0303**] audio track: -a 1 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[**0304**] audio track: -a 2 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[tcprobe] V: 185950 frames, 7438 sec @ 25.000 fps

[tcprobe] A: 116.22 MB @ 128 kbps

[tcprobe] CD: 650 MB|V: 533.8 MB @ 602.0 kbps

[tcprobe] CD: 700 MB|V: 583.8 MB @ 658.4 kbps

[tcprobe] CD: 1300 MB|V: 1183.8 MB @ 1335.1 kbps

[tcprobe] CD: 1400 MB|V: 1283.8 MB @ 1447.9 kbps

[**0305**] This information is returned by tcprobe, which is part of transcode. Part of the extraction configuration process of S**1** involves determining the configuration of the target device, which is represented by the information stored in the production configuration data area **23**. It is helpful therefore to understand the information that is stored there.

[**0306**] Information data stored in the production configuration data area **23** identifies the aspect ratio of the display of the target device. In most cases, the aspect ratio 4:3, although this may vary form device to device. Certain devices will include 16:9 (widescreen) aspect ratios, although in practice the aspect ratio may take a value which is not the same as a conventional television aspect ratio. The production configuration data also identifies the audio language required. It also identifies whether or not subtitles are required. If they are required, the production information configuration identifies the language that the subtitles are required to be in. The bitrates of the video and the audio tracks are included in the production configuration data. The bitrates may depend on the capabilities of the target device, on the particular media player installed in the target device or on any other factors. The production configuration data may also indicate a maximum volume size, for example indicating the amount of usable memory in an MMC. The production configuration information also includes an indication of the format on which the movie data is to be stored. For example, this format can be 3GPP or MPEG-4 format, or any other suitable format.

[**0307**] The information included in the production configuration data area **23** also includes the type of the target device. This may be, for example, a model number of the mobile telephone on which the movie is to be reproduced. In some circumstances, it may be possible that two different mobile telephones having the same model number can have different hardware and/or software configurations. Where different configurations are possible, and this may have a bearing on the optimum processing effected by the apparatus **10**, the information stored in the production configuration data area **23** preferably also includes details of how the hardware and/or software configuration departs from the standard configuration, or perhaps instead merely specifies the configuration.

[**0308**] The automated extraction configuration module **16** determines from the information returned by tcprobe, (in particular the first line thereof reproduced above) that the DVD **15** contains only widescreen (that is 16:9 aspect ratio) video in MPEG 2 PAL format. The module **16** also determines that there are three audio tracks, identified by the second and fourth lines respectively. The first and second tracks have 6 channels each and 48 kHz sampling rates. The first is in the English language and the second is in the German language, as identified by the "en" and "de" designations. The third audio track is in the English language and is a stereo (two channel) signal having a 48 kHz sampling rate. The module determines also that the DVD **15** has eight subtitle tracks, in various languages. The module **16** also determines the frame rate, the number of frames and the length of the movie. The module **16** uses the last four lines of the returned information to determine the content bitrate variations that can be extracted from the DVD **16**.

[0309] The function of the automated extraction configuration module **16** also includes obtaining decryption keys, which are needed to allow the audio-visual content on the DVD to be reproduced.

[0310] The information determined by the automated extraction configuration module **16** constitutes the configuration of the DVD **15**.

[0311] Based on the information in the production configuration data area **23** and on the DVD configuration information, the automated extraction configuration module **16** makes a decision as to which audio tracks, which video channel (if there is more than one video channel) and which subtitle track are needed. Typically, the subtitle track identified by this process is the first listed subtitle track which is in the same language as the subtitle language identified in the production configuration data area **23**. Also, the audio track identified by this process is the audio track which is in the same language as the audio language identified in the production configuration data area **23** and which is most suitable for use by the target device. In most cases, and Dolby™ Pro Logic™ audio channels will not be suitable, because most target devices will not be equipped to handle such audio signals. A stereo audio track will in most cases be the most suitable audio track, although any mono track may be most suitable for a target device with only mono audio capabilities. The video channel selected by this process typically is the main channel, i.e. the actual movie, and not any 'additional features', such as trailers and behind-the-scene documentaries and the like that are commonly included on DVDs. Data identifying the tracks and channels identified by this process is stored in the extraction configuration data area **17**.

[0312] In step S**2**, the data stored on the DVD **15** is read as a stream. This is represented by the arrow between the movie on DVD data area **15** and the DVD decryption and extraction module **18** in FIG. **1**. It is only the content which is read at this time, since the configuration information, or metadata, is not used by the DVD decryption and extraction module **18** directly. Also, it is only the relevant content which is read. The relevant content is identified to the DVD decryption and extraction module **18** by the information stored in the extraction configuration data area **17**, which identifies the relevant video channel, the relevant audio channel and any relevant subtitle channel. At step S**3**, the relevant portions of the DVD data stream are decrypted by the DVD decryption and extraction module **18**. This decryption uses transcode with the keys extracted by the automated extraction configuration module **16**. Decryption is performed "on the fly", i.e. as a continuous process as the content is read from the DVD **15**. As the data is decrypted, it is converted into the intermediate format, i.e. AVI format. At step S**5**, the movie data is written into the extracted movie data buffer **14** as a file or series of files in the intermediate format.

[0313] At step S**6**, extraction post-processing is performed. This involves splitting or joining the content file or files present in the extracted movie data buffer **14** into components. Whether there is any splitting or any joining and the extent of it depends on the target device configuration information stored in the production configuration data area **23**. In most cases, this step will involve splitting the extracted content cleanly to multiple volumes. Providing movie content in the form of multiple volumes is desirable in many circumstances due to the limitations of mobile telephones. It is a fairly straightforward procedure to split DVD movie content into volumes corresponding to the DVD chapters present on the original DVD **15**. Following step S**6**, the extraction of the movie data is complete.

[0314] The result is movie data stored in the extracted movie data buffer **14** which is encoded into an intermediate format (e.g. AVI format) and which includes only one audio track, which is in the required language identified by the production configuration information stored in production configuration data area **23**, and optionally one subtitled track, in the required language. The extracted movie data typically is divided into a number of volumes, although this may not be necessary depending on the configuration of the target device.

[0315] Instead of using a DVD data source **15**, the other movie data storage area **12** may be used. In this case, format conversion to the intermediate format, for example AVI, is carried out by the format conversion module **13**. If only DVD sources **15** will be used, then the second content source **9** can be omitted. If included, the format conversion module **13** takes a form which is suitable for the particular type of content provided at the other movie data storage area **12**. A separate format conversion module **13** may be needed for each type of data that can be stored in the other movie data storage area **12**.

[0316] The procedure of FIG. **3** begins with the extraction process complete. At step S**1**, the extraction file is read. This is an "on the fly" procedure and is represented by the arrow linking the extracted movie data buffer **14** with the mobile format conversion module **19**. At step S**2**, the mobile format conversion module **19** decodes the content comprising the movie data. The step uses transcode. At step S**3**, the decoded content is encoded into the required mobile format, as identified by the production configuration information stored in the production configuration data area **23**. The encoding is performed by the codec **22**. The encoding is performed in such a way as to result in audio and video content having the most appropriate bitrates. What are the most appropriate bitrates is determined by the mobile format conversion module **19**. In particular, the mobile format conversion module **19** uses knowledge of the number of video frames in the video data and the length of the audio track along with the maximum volume size information stored in the production configuration data area **23** to determine the most suitable bitrates. In most cases, the most suitable bitrates for the audio and video will be the bitrates which are the maximum possible bitrates which could be used to fit the entire content within the maximum volume size.

[0317] Usually, the bitrates selected for the audio and the video give rise to comparable quality for those components, although there can be some discrepancy if this results in mobile format movie data which would give an improved playback experience if this is possible having regard to the maximum volume size. For example, if audio and video content at a certain quality level would give rise to data exceeding the maximum volume size but that content at a quality level immediately below that would give rise to a significant shortfall of the volume size, the mobile format conversion module **19** may make a decision to use the higher

bitrate for the video content and the lower bitrate for the audio content, so as to make the best use of the available volume size.

[0318] If examination of the information stored in the production configuration data area **23** reveals that the target device is not optimised for video playback at the same frame rate as that of the DVD source **15**, then this is taken into account by the mobile format conversion module **19**. In particular, the mobile format conversion module **19** may modify the frame rate of the content data so that it is optimised for the target mobile device. Typically, this will involve a reduction in the frame rate which, because of the limited display size in most mobile telephones, would not be so noticeable as it would if a full size display were used. If the optimal frame rate is not equal to the source frame rate divided by an integer, then the mobile format conversion module **19** may use frame interleaving to effect a smooth result in the generated movie content when played back on a mobile telephone.

[0319] Step S**3** thus utilises information stored in the production configuration data area **23** to control the mobile format conversion module **19** to encode the data using the codec **22** into the appropriate data format and with appropriate bitrates.

[0320] The production configuration data area **23** may be updatable according to the target device which is of interest in a particular format conversion process. In this case, the production configuration data area **23** will store data for only one target device at a time, and this data is changed as required. Alternatively, the production configuration data area **23** stores a set of data for each of plural target devices, and one of the data sets is selected according to the particular target device of interest at a given time. In either case, the apparatus **10** is easily controlled to carry out a format conversion process which is optimised for each of plural target device configurations.

[0321] Digital rights management content is added to step S**4**. This is implemented by the mobile format conversion module **19** using the DRM processing module **21**. The procedure implemented by the step S**4** depends on the target format identified by the information stored in the production configuration data area **23**. What form of DRM content is added may depend in particular on the form of the codec **22**. The form of the codec **22** in turn has an effect on the form of the codec in the media player. In particular, when the codec **22** is a custom codec, a custom form of DRM is used. Here, the form of DRM can be selected to provide optimal operation with the custom media player. If an off-the-shelf codec, such as Real Media™, is used as the codec **22**, a suitable DRM will be used.

[0322] Assuming it is allowed by the media player and the target device, the DRM content may impose content reproduction and distribution restrictions as follows. One option is to limit viewing of the content to the particular target device or user, as for example identified by an IMEI or an IMSI number or any other unique or quasi-unique serial number. In this case, the serial number needs to be included in the production configuration data area **23**, so that the mobile format conversion module **19** can operate with the DRM processing module **21** and the production configuration data area **23** to include suitable DRM content in the movie data. Another option is to allow the movie to be viewable up until a particular time and/or date. Thus, the resulting movie will have a "shelf-life" and will not be viewable after the date and/or time specified by the DRM content. A third option is to allow the movie content to be viewable on a predetermined number of occasions (N times). Once the movie has been viewed N times, the media player in the target device will not allow the content to be refused again, thereby rendering it useless. Alternatively, the media player may be arranged to erase the MMC or otherwise delete or corrupt the movie data immediately after the Nth viewing. Alternatively or in addition, the DRM content can prevent the content being copied or forwarded if not authorised. Thus, it can be said that the DRM content prevents or deters the consumption of the content on mobile devices other than the one for which it was intended and/or copying of the content.

[0323] Preferably, the DRM content is encrypted and included in the header of the resulting movie data, although the DRM content may be included in the movie data in any suitable way. Clearly, if a standard DRM process is required to be used by the target device, the DRM content included in the movie data by the mobile format conversion module **19** in the DRM processing module **21** will conform to the relevant standard.

[0324] At step S**5**, the target content is written to the mobile format movie data area **20** as a file. The file may be an area of memory in a computer server, for instance, or the content file may be written directly onto an MMC or other portable transferable media. The file written by this step S**5** includes content in the appropriate format, and also DRM content either embedded into the movie content or else in a separate file. After step S**5**, the conversion is complete, the result is stored in the mobile format movie data area **20** data constituting the movie originally on the DVD data area **15** but encoded in a format suitable for use by the target mobile device and having appropriate audio and video content bitrates. Furthermore, the movie includes suitable DRM content, multiple volumes if appropriate to the format of the target device, a single audio sound track, and optionally a single subtitle track.

[0325] Where the video content on the DVD **15** has a different aspect ratio to the display of the target device, there preferably is modification of the video signal from the DVD such that it corresponds to the aspect ratio of the target device. This can be carried out by the DVD encryption and extraction module **18**. Preferably however, modification of the video signal from the DVD **15** such that it corresponds to the aspect ratio of the target device is carried out by the mobile format conversion module **19**. The modification may involve simple cropping from the left and right sides of images if narrower images are required, or cropping from the top and bottom of images if wider images are required. The modification may involve as well or instead a limited amount of image stretching, either widthwise or heightwise. In this case, it is preferred to have more picture linearity in the central region of the display than at the edges of the display. Thus, compression or stretching is effected to a greater degree at the edges of the images than it is a central portion. The DVD encryption and extraction module **18** or the mobile format conversion module **19**, as the case may be, can be pre-programmed to make a decision as to what cropping and/or stretching is required on the basis of a

look-up table relating course aspect ratios to target device aspect ratios and the corresponding modification required, or in any other suitable way.

[0326] In accordance with the invention, the data written to the mobile format movie data area **20** also includes two or more media players. This is advantageous for a number of reasons. Firstly, it reduces the number of factors which need to be taken into account by the mobile format conversion module **19**. The target device configuration information does not need to include information identifying the media player included in the mobile device, since this is not needed when the media player is included with the movie content data. Secondly, it allows movie content data to be consumed even if no suitable media player, or indeed no media player at all, is included in the mobile device.

[0327] The media player or players may be embedded, or alternatively included alongside, the movie content data. Embedding the media player into the content data allows easier control of the movie content, and makes it very difficult for the movie content data to be separated by unauthorised persons. Each media player typically consumes less than 1 MB of memory.

[0328] In one embodiment, a number of different media players are stored, along with the movie content data and a loader program. The mobile device is controlled to run the loader program initially. The program detects the relevant configuration of the mobile device and determines therefrom which of the media players to use to consume the movie content data. In this way, it is possible to utilise an MMC card for a greater number of target device configurations, which clearly can be advantageous, especially when the MMC cards are intended for retail from a shop display or similar.

[0329] The loader program preferably is arranged to control the mobile device to detect whether or not it already includes a suitable media player. If a suitable media player is detected, this is controlled to be used instead of installing a media player from the MMC card onto the mobile device. This is advantageous since it reduces the possibility of there being an installation or deinstallation error, thereby improving the reliability of the mobile device.

[0330] In a second embodiment, instead of including multiple separable media players, multiple media players may be provided through a single configurable media player software application. In this case, the loader program may determine what media player is required, and operate appropriate software modules forming part of the media player software. Software module or functions which are not appropriate for the mobile device configuration are not used. Thus, multiple media players are made up from a single software application, which reuses modules or functions for certain media player functionality. Where a single media player software application is used, the loader program may form part of the media player software application itself.

[0331] The movie content data, as well as any media player(s), stored in the mobile format movie data area **20** can be communicated to the target mobile device in any suitable way. For the next few years at least, it is envisaged that mostly MMC or other transferable media will be used to store and transport the movie content. However, as mobile data transfer becomes faster and cheaper, it is expected that movie content will be transferable over-the-air, for example using WAP or 3G data transfer. Transfer may instead be effected by transfer from an Internet connected PC which has downloaded the movie content from a website, using a short range link such as a cable, or wirelessly using IrDA or Bluetooth, or using a transferable storage medium such as an MMC.

[0332] A mobile device is shown schematically in FIG. **4**. Here, the mobile telephone **40** includes all the conventional components needed for voice communication, although these are not shown for the sake of clarity. The telephone **40** includes a movie decode module **41**, which is in bidirectional communication with a codec **42**. A movie is stored in a mobile movie data area **43**, which may take any suitable form. It may for example be an MMC, a memory space connected by way of an external drive, internal RAM or other memory, or it may take any other suitable form. A DRM validation module **44** is connected to receive DRM data from the data in the mobile movie data area **43**. The DRM validation module **44** controls the movie decoder module **41** to allow or disallow it to decode the movie data from the mobile movie data area **43** on the basis of a decision made using the DRM data, and time/date or serial number inputs as appropriate. When allowed by the DRM validation module **44** to decode movie data from the mobile movie data area **43** and when controlled to do so by user input, the movie decoder module **41** uses the codec **42** to decode the data and provide decoded data to a buffer **45**. From the buffer **45**, the movie is displayed on a display **46** by a display module **47**. The display module provides control data to the movie decoder module **41** so as to enable decoding at a suitable rate.

[0333] The mobile telephone **40** is arranged to install a loader program from the mobile movie data area **43**, if one is stored there. The loader program then causes the mobile telephone **40** to determine its configuration, and to select a media player, also stored in the mobile movie data area **43**, accordingly. This media player then is used to consume the movie content data. If a suitable media player is already installed in the mobile telephone **40**, then this is used instead, and no media player then is installed from the mobile movie data area **43**.

[0334] Although the mobile device is said to be a mobile (cellular) telephone, it may instead be a personal digital assistant (PDA), which may or may not have bidirectional voice communication capabilities. The invention is primarily concerned with providing audio-visual content on a device which is designed primarily for another function. However, the invention is concerned also with dedicated media players.

[0335] Where a movie on a DVD is to be provided onto transferable media for use with a general class of target mobile devices, or even where the movie is to be provided for more than a small number of target devices on the same model number, a system such as a system shown in FIG. **5** can be used to advantage. In FIG. **5**, first to third servers **30**, **31**, **32** are shown. The first server **30** is designated as a management node, and includes connections to each of the second and third servers **31**, **32**, which constitute child nodes. Each of the servers **30** to **32** includes at least first and second DVD drives **33**. In this example, DVDs need to be inserted into and extracted from the DVD drives **33** manu-

ally, although it is possible to use robots or other automation for this task instead if required.

[0336] Each of the servers **30** to **32** extracts and converts films from DVDs in the DVD drives **33** in parallel. Movies can be extracted from DVDs in a single drive sequentially, i.e. one after the other.

[0337] Assuming sufficient speed for the DVD drive **33** and sufficient processing speed for the servers **30** to **32**, the DVD extraction and conversion process can be completed in respect of one DVD in tens of minutes. Thus, where a serial number of a target device. or similar is to be included in the resulting movie to enable the movie to be reproduced only on that target device, the conversion process needs to be effected once for each specific target device. It will be appreciated that the extraction process needs to be performed only once, since the extracted movie is stored in the extracted movie data buffer.

[0338] Where a movie is to be used for a number of target devices of the same class, then the extraction and conversion processes need to be performed only once. Once the movie is stored in mobile format in the mobile format movie data area **20**, it can be copied to an MMC or other removable media device as many times is required. This can be carried out in a suitable manner, for example using internal or external MMC drives.

[0339] The setup for the management system installation specific architecture is in flat files, for example, in a /etc/ subdirectory. The setup for movie production is in database tables using a custom Postgres or Oracle database, although any other suitable database can be used instead, depending on the scale and performance requirements. The management system running on the child node servers **31**, **32** communicate with the management system on the first server **30**. The management node **30** is responsible for task allocation. One instance of the management system is required for each conversion session.

Storing Content

[0340] The invention still further relates to a portable data storage medium including a security device.

[0341] There is a trend for mobile telephones, also known as cellular telephones, to be provided with colour displays having many thousands of pixels. As time progresses the quality of these displays and the resolutions afforded thereby increases. Furthermore, semiconductor terminology is such the mobile telephones can be provided with quite substantial amounts of memory. Whereas previously it has been known to incorporate MP3 players and the like into mobile telephones, the provision of improved displays and increased amounts of memory allows mobile telephones to be used for use as limited digital television receivers. It has been proposed as well to provide audio-visual content on a multimedia card (MMC), for viewing on a mobile telephone. The Nokia™ 7610 is one such capable mobile telephone. This telephone can handle 3GPP and RealMedia audio-visual formats.

[0342] The provision of copyright works onto MMCs for sale to the public potentially provides an opportunity for the content to be illegally copied and distributed. The invention aims to provide a memory device, such as an MMC, in which the content cannot be easily accessed in such a way as to allow it to be copied but which can allow it to be played out for consumption.

[0343] According to a still further aspect of the invention, there is provided a portable data storage medium comprising:

[0344] non-volatile memory having stored therein data comprising computer-readable instructions constituting a media player

[0345] an interface including terminals for connecting to an external device;

[0346] a controller operable to read data out from the non-volatile memory and feed it to the interface; and

[0347] a security device,

in which the media player is operable when running on an external device to interact with the security device in such a way that the security device can determine whether the media player is entitled to access content data from the non-volatile memory, the security device allowing or disallowing access to the content data accordingly.

[0348] A data terminal of the controller can be connected to the interface via the security device.

[0349] Alternatively, the security device can be integral with the controller. In this case, the controller and security device may be operable to decrypt content data read out from the non-volatile memory.

[0350] Advantageously, the controller is operable also to write data from the interface to the non-volatile memory. In this case, if the controller and security device is operable to decrypt content data read out from the non-volatile memory, it may be operable also to encrypt data written from the interface to the non-volatile memory.

[0351] Embodiments of the still further aspect of the present invention will now be described by way of example only, with reference to the accompanying drawings in which:

[0352] FIG. **1** is a schematic diagram of audio-visual content provision apparatus;

[0353] FIGS. **2** and **3** are flowcharts illustrating steps of operation of the FIG. **1** apparatus;

[0354] FIG. **4** is a schematic drawing illustrating apparatus for playback of the converted audio-visual content in a mobile telephone;

[0355] FIG. **5** illustrates a combination of an MMC and the mobile telephone of FIG. **4**;

[0356] FIGS. **6** and **7** illustrate alternative embodiments of MMC hardware, according to the invention;

[0357] FIG. **8** is a flowchart illustrating security validation between the mobile telephone of FIG. **4** and the MMC of FIG. **6** or FIG. **7**; and

[0358] FIG. **15** is a schematic drawing of a system of interconnected computers.

[0359] Throughout the drawings, reference numerals are re-used for like elements.

[0360] Referring firstly to FIG. **1**, content extracting and converting apparatus **10** is illustrated schematically. Two alternative sources of audio-visual content **8**, **9** are included. A first content source **8** utilises film or movie data stored on a DVD (digital video disk or digital versatile disk) **15**. An automated extraction configuration module **16** examines metadata stored on the DVD **15** to determine the configuration of content data stored on the DVD. This involves the application of a tcprobe, and an analysis of the information returned from the DVD **15**. This is described in more detail below. The result is data stored in an extraction configuration memory area **17** representing an extraction configuration. The extraction configuration data from the memory area **17** is utilised by a DVD decryption and extraction module **18** to extract movie data (i.e. the content data) from the DVD **15**. The result is content data in an intermediate format, which is written to an intermediate format movie data area **14**. The data included in the intermediate format movie data area **14** is in predetermined format and is suitable for conversion into a form ready for reproduction on a mobile telephone (not shown). Preferably the intermediate format is AVI. This format has the advantage of high resolution, yet is relatively easy to handle and it is relatively easy to convert from AVI into 3GPP and many other formats suitable for use by mobile devices.

[0361] The second source of audio-visual content **9** receives from a movie data storage area **12** data representing a movie (or film) in AVI (audio-visual interleave) or other format. The movie so supplied is converted by a format conversion module **13** before being written to the intermediate format movie data area **14**.

[0362] Thus, either of the audio-visual content sources **8**, **9** can be used to provide movie data in the intermediate format movie data area **14**.

[0363] A mobile format conversion module **19** converts movie data stored in the extracted movie data area **14** and provides a movie in mobile telephone consumable format in a mobile format movie data area **20**. The mobile format conversion module **19** utilises a digital rights management (DRM) processing module **21**, which allows certain control over the access and distribution of the resulting movie data. The conversion effected by the mobile format conversion module **19** uses a codec **22**, which preferably is custom-designed for the purpose. Importantly, the conversion effected by the mobile format conversion module **19** uses information stored in a production configuration data area **23**. By controlling the mobile format conversion module **19** on the basis of information specific to the configuration of, and thus tailored to, a target device, the apparatus **10** can be used to provide movie data for any of potentially a large number of target mobile devices.

[0364] The extraction effected by the audio-visual content source **12** will now be described in detail with reference to FIG. **2**.

[0365] In FIG. **2**, extraction configuration is effected at step S**1**. This utilises the automated extraction configuration **16** shown in FIG. **1**. Extraction configuration commences by analysing the DVD source **15**. The result of an example analysis, i.e. what is returned in response to a query, is illustrated below:

(dvd_reader.c) mpeg2 pal 16:9 only letterboxed U0 720x576 video

(dvd_reader.c) ac3 en drc 48 kHz 6Ch

(dvd_reader.c) ac3 de drc 48 kHz 6Ch

(dvd_reader.c) ac3 en drc 48 kHz 2Ch

(dvd_reader.c) subtitle 00=<en>

(dvd_reader.c) subtitle 01=<de>

(dvd_reader.c) subtitle 02=<sv>

(dvd_reader.c) subtitle 03=<no>

(dvd_reader.c) subtitle 04=<da>

(dvd_reader.c) subtitle 05=<fi>

(dvd_reader.c) subtitle 06=<is >

(dvd_reader.c) subtitle 07=<en>

(dvd_reader.c) subtitle 08=<de>

[tcprobe] summary for /media/dvdrecorder/, (*)=not default, 0=not detected

import frame size: -g 720x576 [720x576]

[0366] aspect ratio: 16:9 (*)

[0367] frame rate: -f 25.000 [25.000] frc=3

[0368] audio track: -a 0 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[0369] audio track: -a 1 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[0370] audio track: -a 2 [0]-e 48000,16,2 [48000,16,2]-n 0x2000 [0x2000]

[tcprobe] V: 185950 frames, 7438 sec @ 25.000 fps

[tcprobe] A: 116.22 MB @ 128 kbps

[tcprobe] CD: 650 MB|V: 533.8 MB @ 602.0 kbps

[tcprobe] CD: 700 MB|V: 583.8 MB @ 658.4 kbps

[tcprobe] CD: 1300 MB|V: 1183.8 MB @ 1335.1 kbps

[tcprobe] CD: 1400 MB|V: 1283.8 MB @ 1447.9 kbps

[0371] This information is returned by tcprobe, which is part of transcode.

[0372] Part of the extraction configuration process of S**1** involves determining the configuration of the target device, which is represented by the information stored in the production configuration data area **23**. It is helpful therefore to understand the information that is stored there.

[0373] Information data stored in the production configuration data area **23** identifies the aspect ratio of the display of the target device. In most cases, the aspect ratio 4:3, although this may vary form device to device. Certain devices will include 16:9 (widescreen) aspect ratios, although in practice the aspect ratio may take a value which is not the same as a conventional television aspect ratio. The production configuration data also identifies the audio language required. It also identifies whether or not subtitles are required. If they are required, the production information configuration identifies the language that the subtitles are required to be in. The bitrates of the video and the audio

tracks are included in the production configuration data. The bitrates may depend on the capabilities of the target device, on the particular media player installed in the target device or on any other factors. The production configuration data may also indicate a maximum volume size, for example indicating the amount of usable memory in an MMC. The production configuration information also includes an indication of the format on which the movie data is to be stored. For example, this format can be 3GPP or MPEG-4 format, or any other suitable format.

[0374] The information included in the production configuration data area **23** also includes the type of the target device. This may be, for example, a model number of the mobile telephone on which the movie is to be reproduced. In some circumstances, it may be possible that two different mobile telephones having the same model number can have different hardware and/or software configurations. Where different configurations are possible, and this may have a bearing on the optimum processing effected by the apparatus **10**, the information stored in the production configuration data area **23** preferably also includes details of how the hardware and/or software configuration departs from the standard configuration, or perhaps instead merely specifies the configuration.

[0375] The automated extraction configuration module **16** determines from the information returned by tcprobe, (in particular the first line thereof reproduced above) that the DVD **15** contains only widescreen (that is 16:9 aspect ratio) video in MPEG 2 PAL format. The module **16** also determines that there are three audio tracks, identified by the second and fourth lines respectively. The first and second tracks have 6 channels each and 48 kHz sampling rates. The first is in the English language and the second is in the German language, as identified by the "en" and "de" designations. The third audio track is in the English language and is a stereo (two channel) signal having a 48 kHz sampling rate. The module determines also that the DVD **15** has eight subtitle tracks, in various languages. The module **16** also determines the frame rate, the number of frames and the length of the movie. The module **16** uses the last four lines of the returned information to determine the content bitrate variations that can be extracted from the DVD **16**.

[0376] The function of the automated extraction configuration module **16** also includes obtaining decryption keys, which are needed to allow the audio-visual content on the DVD to be reproduced.

[0377] The information determined by the automated extraction configuration module **16** constitutes the configuration of the DVD **15**.

[0378] Based on the information in the production configuration data area **23** and on the DVD configuration information, the automated extraction configuration module **16** makes a decision as to which audio tracks, which video channel (if there is more than one video channel) and which subtitle track are needed. Typically, the subtitle track identified by this process is the first listed subtitle track which is in the same language as the subtitle language identified in the production configuration data area **23**. Also, the audio track identified by this process is the audio track which is in the same language as the audio language identified in the production configuration data area **23** and which is most suitable for use by the target device. In most cases, Dolby™ Pro Logic™ audio channels will not be suitable, because most target devices will not be equipped to handle such audio signals. A stereo audio track will in most cases be the most suitable audio track, although any mono track may be most suitable for a target device with only mono audio capabilities. The video channel selected by this process typically is the main channel, i.e. the actual movie, and not any 'additional features', such as trailers and behind-the-scene documentaries and the like that are commonly included on DVDs. Data identifying the tracks and channels identified by this process is stored in the extraction configuration data area **17**.

[0379] In step S**2**, the data stored on the DVD **15** is read as a stream. This is represented by the arrow between the movie on DVD data area **15** and the DVD decryption and extraction module **18** in FIG. **1**. It is only the content which is read at this time, since the configuration information, or metadata, is not used by the DVD decryption and extraction module **18** directly. Also, it is only the relevant content which is read. The relevant content is identified to the DVD decryption and extraction module **18** by the information stored in the extraction configuration data area **17**, which identifies the relevant video channel, the relevant audio channel and any relevant subtitle channel. At step S**3**, the relevant portions of the DVD data stream are decrypted by the DVD decryption and extraction module **18**. This decryption uses transcode with the keys extracted by the automated extraction configuration module **16**. Decryption is performed "on the fly", i.e. as a continuous process as the content is read from the DVD **15**. As the data is decrypted, it is converted into the intermediate format, i.e. AVI format. At step S**5**, the movie data is written into the extracted movie data buffer **14** as a file or series of files in the intermediate format.

[0380] At step S**6**, extraction post-processing is performed. This involves splitting or joining the content file or files present in the extracted movie data buffer **14** into components. Whether there is any splitting or any joining and the extent of it depends on the target device configuration information stored in the production configuration data area **23**. In most cases, this step will involve splitting the extracted content cleanly to multiple volumes. Providing movie content in the form of multiple volumes is desirable in many circumstances due to the limitations of mobile telephones. It is a fairly straightforward procedure to split DVD movie content into volumes corresponding to the DVD chapters present on the original DVD **15**. Following step S**6**, the extraction of the movie data is complete.

[0381] The result is movie data stored in the extracted movie data buffer **14** which is encoded into an intermediate format (e.g. AVI format) and which includes only one audio track, which is in the required language identified by the production configuration information stored in production configuration data area **23**, and optionally one subtitled track, in the required language. The extracted movie data typically is divided into a number of volumes, although this may not be necessary depending on the configuration of the target device.

[0382] Instead of using a DVD data source **15**, the other movie data storage area **12** may be used. In this case, format conversion to the intermediate format, for example AVI, is carried out by the format conversion module **13**. If only

DVD sources **15** will be used, then the second content source **9** can be omitted. If included, the format conversion module **13** takes a form which is suitable for the particular type of content provided at the other movie data storage area **12**. A separate format conversion module **13** may be needed for each type of data that can be stored in the other movie data storage area **12**.

[**0383**] The procedure of FIG. **3** begins with the extraction process complete. At step S**1**, the extraction file is read. This is an "on the fly" procedure and is represented by the arrow linking the extracted movie data buffer **14** with the mobile format conversion module **19**. At step S**2**, the mobile format conversion module **19** decodes the content comprising the movie data. The step uses transcode. At step S**3**, the decoded content is encoded into the required mobile format, as identified by the production configuration information stored in the production configuration data area **23**. The encoding is performed by the codec **22**. The encoding is performed in such a way as to result in audio and video content having the most appropriate bitrates. What are the most appropriate bitrates is determined by the mobile format conversion module **19**. In particular, the mobile format conversion module **19** uses knowledge of the number of video frames in the video data and the length of the audio track along with the maximum volume size information stored in the production configuration data area **23** to determine the most suitable bitrates. In most cases, the most suitable bitrates for the audio and video will be the bitrates which are the maximum possible bitrates which could be used to fit the entire content within the maximum volume size.

[**0384**] Usually, the bitrates selected for the audio and the video give rise to comparable quality for those components, although there can be some discrepancy if this results in mobile format movie data which would give an improved playback experience if this is possible having regard to the maximum volume size. For example, if audio and video content at a certain quality level would give rise to data exceeding the maximum volume size but that content at a quality level immediately below that would give rise to a significant shortfall of the volume size, the mobile format conversion module **19** may make a decision to use the higher bitrate for the video content and the lower bitrate for the audio content, so as to make the best use of the available volume size.

[**0385**] If examination of the information stored in the production configuration data area **23** reveals that the target device is not optimised for video playback at the same frame rate as that of the DVD source **15**, then this is taken into account by the mobile format conversion module **19**. In particular, the mobile format conversion module **19** may modify the frame rate of the content data so that it is optimised for the target mobile device. Typically, this will involve a reduction in the frame rate which, because of the limited display size in most mobile telephones, would not be so noticeable as it would if a full size display were used. If the optimal frame rate is not equal to the source frame rate divided by an integer, then the mobile format conversion module **19** may use frame interleaving to effect a smooth result in the generated movie content when played back on a mobile telephone.

[**0386**] Step S**3** thus utilises information stored in the production configuration data area **23** to control the mobile format conversion module **19** to encode the data using the codec **22** into the appropriate data format and with appropriate bitrates.

[**0387**] The production configuration data area **23** may be updatable according to the target device which is of interest in a particular format conversion process. In this case, the production configuration data area **23** will store data for only one target device at a time, and this data is changed as required. Alternatively, the production configuration data area **23** stores a set of data for each of plural target devices, and one of the data sets is selected according to the particular target device of interest at a given time. In either case, the apparatus **10** is easily controlled to carry out a format conversion process which is optimised for each of plural target device configurations.

[**0388**] Digital rights management content is added to step S**4**. This is implemented by the mobile format conversion module **19** using the DRM processing module **21**. The procedure implemented by the step S**4** depends on the target format identified by the information stored in the production configuration data area **23**. What form of DRM content is added may depend in particular on the form of the codec **22**. The form of the codec **22** in turn has an effect on the form of the codec in the media player. In particular, when the codec **22** is a custom codec, a custom form of DRM is used. Here, the form of DRM can be selected to provide optimal operation with the custom media player. If an off-the-shelf codec, such as Real Media m, is used as the codec **22**, a suitable DRM will be used.

[**0389**] Assuming it is allowed by the media player and the target device, the DRM content may impose content reproduction and distribution restrictions as follows. One option is to limit viewing of the content to the particular target device or user, as for example identified by an IMEI or an IMSI number or any other unique or quasi-unique serial number. In this case, the serial number needs to be included in the production configuration data area **23**, so that the mobile format conversion module **19** can operate with the DRM processing module **21** and the production configuration data area **23** to include suitable DRM content in the movie data. Another option is to allow the movie to be viewable up until a particular time and/or date. Thus, the resulting movie will have a "shelf-life" and will not be viewable after the date and/or time specified by the DRM content. A third option is to allow the movie content to be viewable on a predetermined number of occasions (N times). Once the movie has been viewed N times, the media player in the target device will not allow the content to be refused again, thereby rendering it useless. Alternatively, the media player may be arranged to erase the MMC or otherwise delete or corrupt the movie data immediately after the Nth viewing. Alternatively or in addition, the DRM content can prevent the content being copied or forwarded if not authorised. Thus, it can be said that the DRM content prevents or deters the consumption of the content on mobile devices other than the one for which it was intended and/or copying of the content.

[**0390**] Preferably, the DRM content is encrypted and included in the header of the resulting movie data, although the DRM content may be included in the movie data in any suitable way. Clearly, if a standard DRM process is required to be used by the target device, the DRM content included

in the movie data by the mobile format conversion module **19** in the DRM processing module **21** will conform to the relevant standard.

[**0391**] At step S**5**, the target content is written to the mobile format movie data area **20** as a file. The file may be an area of memory in a computer server, for instance, or the content file may be written directly onto an MMC or other portable transferable media. The file written by this step S**5** includes content in the appropriate format, and also DRM content either embedded into the movie content or else in a separate file. After step S**5**, the conversion is complete, the result is stored in the mobile format movie data area **20** data constituting the movie originally on the DVD data area **15** but encoded in a format suitable for use by the target mobile device and having appropriate audio and video content bitrates. Furthermore, the movie includes suitable DRM content, multiple volumes if appropriate to the format of the target device, a single audio sound track, and optionally a single subtitle track.

[**0392**] Where the video content on the DVD **15** has a different aspect ratio to the display of the target device, there preferably is modification of the video signal from the DVD such that it corresponds to the aspect ratio of the target device. This can be carried out by the DVD encryption and extraction module **18**. Preferably however, modification of the video signal from the DVD **15** such that it corresponds to the aspect ratio of the target device is carried out by the mobile format conversion module **19**. The modification may involve simple cropping from the left and right sides of images if narrower images are required, or cropping from the top and bottom of images if wider images are required. The modification may involve as well or instead a limited amount of image stretching, either widthwise or heightwise. In this case, it is preferred to have more picture linearity in the central region of the display than at the edges of the display. Thus, compression or stretching is effected to a greater degree at the edges of the images than it is a central portion. The DVD encryption and extraction module **18** or the mobile format conversion module **19**, as the case may be, can be pre-programmed to make a decision as to what cropping and/or stretching is required on the basis of a look-up table relating course aspect ratios to target device aspect ratios and the corresponding modification required, or in any other suitable way.

[**0393**] The data written to the mobile format movie data area **20** also includes one or more media players. This is advantageous for a number of reasons. Firstly, it reduces the number of factors which need to be taken into account by the mobile format conversion module **19**. The target device configuration information does not need to include information identifying the media player included in the mobile device, since this is not needed when the media player is included with the movie content data. Secondly, it allows movie content data to be consumed even if no suitable media player, or indeed no media player at all, is included in the mobile device.

[**0394**] The media player or players may be embedded, or alternatively included alongside, the movie content data. Embedding the media player into the content data allows easier control of the movie content, and makes it very difficult for the movie content data to be separated by unauthorised persons. Each media player typically consumes less than 1 MB of memory.

[**0395**] In one embodiment, a single custom media player is included with the movie content data. After the data is written onto an MMC card, the data relating to the media player is extracted by the mobile device from the MMC and the media player run to process the movie content data.

[**0396**] In another embodiment, a number of different media players are stored, along with the movie content data and a loader program. The mobile device is controlled to run the loader program initially. The program detects the relevant configuration of the mobile device and determines therefrom which of the media players to use to consume the movie content data. In this way, it is possible to utilise an MMC card for a greater number of target device configurations, which clearly can be advantageous, especially when the MMC cards are intended for retail from a shop display or similar.

[**0397**] Instead of including multiple separable media players, multiple media players may be provided through a single configurable media player software application. In this case, the loader program may determine what media player is required, and operate appropriate software modules forming part of the media player software. Software module or functions which are not appropriate for the mobile device configuration are not used. Thus, multiple media players are made up from a single software application, which reuses modules or functions for certain media player functionality. Where a single media player software application is used, the loader program may form part of the media player software application itself.

[**0398**] The movie content data, as well as the media player(s), stored in the mobile format movie data area **20** can be communicated to the target mobile device in an MMC or other transferable media used to store and transport the movie content.

[**0399**] A mobile device is shown schematically in FIG. **4**. Here, the mobile telephone **40** includes all the conventional components needed for voice communication, although these are not shown for the sake of clarity. The telephone **40** includes a movie decoder module **41**, which is in bidirectional communication with a codec **42**. A movie is stored in a mobile movie data area **43**, which may take any suitable form. It may for example be an MMC, a memory space connected by way of an external drive, internal RAM or other memory, or it may take any other suitable form. A DRM validation module **44** is connected to receive DRM data from the data in the mobile movie data area **43**. The DRM validation module **44** controls the movie decoder module **41** to allow or disallow it to decode the movie data from the mobile movie data area **43** on the basis of a decision made using the DRM data, and time/date or serial number inputs as appropriate. When allowed by the DRM validation module **44** to decode movie data from the mobile movie data area **43** and when controlled to do so by user input, the movie decoder module **41** uses the codec **42** to decode the data and provide decoded data to a buffer **45**. From the buffer **45**, the movie is displayed on a display **46** by a display module **47**. The display module provides control data to the movie decoder module **41** so as to enable decoding at a suitable rate.

[**0400**] The mobile telephone **40** may be arranged to install a loader program from the mobile movie data area **43**, if one is stored there. The loader program then causes the mobile

telephone **40** to determine its configuration, and to select a media player, which is a software application and which is also stored in the mobile movie data area **43**, accordingly. This media player then is used to consume the movie content data. Using a proprietary media player stored in the mobile movie data area **43** can be advantageous since is allows effective control over the security of the content data, and allows other features not necessarily available with off-the-shelf or pre-installed media players.

[**0401**] The combination of an MMC and mobile device is illustrated in FIG. **5**. Here, the mobile device **40** is shown to include a CPU **51**, which provides video signals to the display **46**, via a display driver (not shown), and to an audio output device **52** (e.g. headphone socket or speaker, via an audio device driver (not shown). The CPU **51** is connected via a bus to ROM **53**, to RAM **54** and to an MMC connector and interface **55**. An MMC **56** is connected to the mobile device **40** by the MMC connector and interface **55**.

[**0402**] The MMC has stored in its internal non-volatile memory movie content data **57**, three different media players **58**, and a loader program **59**. When content is required to be played-out from the MMC, the mobile device loads the loader program **59**, which decides which of the media players **58** is most suitable by determining configuration parameters of the mobile device **40** and comparing them to parameters of the media players. This media player then is selected on the basis of the determination, is loaded onto the mobile device **40**, and is run (i.e. the media player program is processed) to reproduce the content from the content data **57**. As is conventional, operation of the media player **58** involves storing the media player program in the RAM **54**, and using the CPU **51** to extract relevant data from the MMC **56**, to decode it and to render the resulting content. FIG. **5** is schematic, and detail not relevant to the invention is omitted.

[**0403**] The or each media player is arranged to detect the properties of the display **46** of the host mobile device **40**. In particular, the media player detects the display dimensions and orientation, in terms of numbers of pixels in height and width. The player is arranged to control reproduction of the video content on the display **46** in an orientation which is most suited to the mobile device **40**. If the display **46** is wider than it is high, then video content is reproduced with conventional orientation, i.e. without its orientation being modified. If however the display **46** is determined to be higher than it is wide, the media player reproduces the video content rotated by 90 degrees. Thus, the media player ensures that the video content always is reproduced in landscape format (wider than tall) regardless of screen dimensions. This allows more effective use of the area of the display **46**.

[**0404**] When the video content is rotated on a display **46** by the media player, the functions of a number of keys on a keypad (not shown) or other input device are caused by the media player **40** to be modified so as to be different to their functions when the video content is not rotated by the media player. Since the mobile device **40** will need to be rotated onto its side before the video can be viewed in its intended orientation, providing different key functions with different orientations allows the same control experience to be provided to a user regardless of the orientation of the mobile device **40**. Thus, modifying the controls allows control of the media player using the keypad or other input device to be more convenient and more intuitive for a user. The controls of particular importance are volume up/down, play, pause, forward and rewind, etc.

[**0405**] When the mobile device **40** is not a high specification device, i.e. it has relatively low content handling capability and/or a low resolution display, the media player is arranged such that it can access content from the MMC and not access content from other sources. This allows the content on the MMC to be optimised for reproduction by the proprietary media player, thus providing richer content reproduction than would otherwise be available considering memory size and other technical limitations of the MMC. This feature does not impinge on the ability of the media player to use a standard CODEC **42** pre-existing within the mobile device **40**. Indeed, the media player may utilise standard or other third-party CODECs, or it may utilise a proprietary CODEC.

[**0406**] When being run on higher specification mobile devices **40**, a different media player **58** is used. Here, the media player selected by the loader program **59** is one which is operable to scale non-optimal content for best presentation.

[**0407**] Alternatively, one media player **58** which has adjustable functionality is provided on the MMC **56**. Such a media player does not require a loader program. When running on a mobile device **40**, this media player **58** detects the relevant characteristics of the mobile device **40** and activates appropriate components and functionality of the media player **58** and refrains from activating other components and functionality.

[**0408**] A typical MMC hardware design consists of a flash memory device and a memory/interface controller residing on a very thin PCB (printed circuit board) in a very low profile plastic housing. The underside of the PCB generally forms the bottom of the housing. There are a number of different sizes of MMC.

[**0409**] According to the invention, the MMC hardware is non-conventional, and includes additional security features. A proprietary media player **58** is used to unlock and read content on the secure MMC.

[**0410**] A first embodiment of a novel MMC will now be described with reference to FIG. **6**. Here, an MMC **56** includes a housing **60** in which connector pins **61** are provided. The connector pins form part of a host communications interface to an external device, such as the mobile device **40**. The MMC **56** also includes non-volatile memory **62**, connected to a memory and interface controller **63**, which controls access to the memory **62** and interfaces to the connector pins **61**. The MMC thus far described is conventional. The MMC **56** also includes a security device **64**, which is not conventional. The security device **64** is interposed between the memory and interface controller **63** and the connector pins **61**. Thus, the memory and interface controller **63** and the data (DAT), command (CMD) and clock (CLK) ones of the connector pins **61** are not connected directly since at least some connection between these components is via the security device **64**. VCC, VSS**1** and VSS**2** ones of the connector pins **61** are connected to both the security device **64** and the memory and interface controller **63** in parallel. The security device **64** may be implemented

as a microcontroller, an ASIC (application specific integrated circuit) or an FPGA (field programmable gate array). The components of the MMC **56** are mounted onto a PCB (printed circuit board), which forms part of the housing **60**. Thus, the MMC **56** may have the same dimensions and the same external connectors as a conventional MMC.

[**0411**] The security device **64** is arranged to intercept data and commands communicated between the host device, e.g. the mobile device **40**, and the memory and interface controller **63**. This intercepted data is processed and either is passed through the security device **64** modified or unmodified, or alternatively is replaced by data generated by the security device **64** itself.

[**0412**] Specific data or commands passed in any response can switch the security device **64** into an active mode, in which the security device **64** reads or writes to one of the memory and interface controller **63** and the host interface **61**, masquerading as the other one of those devices. In the active mode, the security device **64** also independently, i.e. without external control, interrogates the memory and interface controller **63** and either prepares data for subsequent host requests or writes data to the non-volatile memory **62** for subsequent requests.

[**0413**] The provision of the active mode allows copy protection to be achieved through cooperation between the MMC **56** and the media player **58**.

[**0414**] The security device **64** does not restrict access to regions of the non-volatile memory **62** where unprotected content resides, in both read and write modes. This allows the MMC **56** including the security device **64** to be used conventionally, i.e. without the security features provided by the security device being operational. The security device **64** can be activated only by authorised entities, such as those licensed to place copyright content, e.g. movies, onto the MMC **56**.

[**0415**] The MMC **56** and the media player **58** are provided with the same serial number. During configuration, the media player **58** is provided also with the result of application of the derail number to a hash function, hereafter termed the hash of the serial number. The memory and interface controller **63** is controlled by the security device **64** to store at programming time (i.e. when it is programmed before sale) the serialised data serial number, a preconfigured security code, and the hash of the serial number.

[**0416**] Validation of the MMC **56** by the media player **58** and validation of the media player **58** by the MMC **56** will now be described with reference to FIG. **8**.

[**0417**] When the MMC **56** with content, one or more media players **58** and optionally a loader program **59** loaded onto it is connected with a mobile device **40**, the media player is made visible in a menu thereof, and thus becomes able to be activated as with any other software application present on the mobile device **40**. When the media player **58** first is started, a first security validation is implemented, in which the following occurs. Firstly, the most suitable media player **58** is uploaded to the mobile device **58**. The media player **58** then at step S**8**.**1** sends the hash of the serial number to the security device **64**. The security device **64** at step S**8**.**2** then compares this with its internally stored hash of the serial number. If the comparison at step S**8**.**3** reveals a match, it is initially assumed that the media player **58** and the MMC **56** are matched, and the security device **64** unlocks the MMC **56** at step S**8**.**4**. The security device **64** then sends at step S**8**.**5** the preconfigured validation code to the media player **58**. Alternatively, if the comparison does not reveal a match, the security device **64** at step S**8**.**6** does not respond. When the media player **58** receives a validation code, it performs at step S**8**.**7** a 32 bit CRC (cyclic redundancy check) calculation on the validation code. On the basis of this calculation, the media player **58** determines at step S**8**.**8** whether the MMC **56** is the one associated with the media player **58**, and unlocks the media player at step S**8**.**9** if appropriate, or else aborts with an error message at step S**8**.**10**. At this stage, the media player **58** can read data from unprotected areas on thereon-volatile memory **62**, if any such areas are present.

[**0418**] A second stage security check is performed when playing the content. After the media controls on the MMC **56** are unlocked and the data becomes readable, data is read out from the non-volatile memory **62** to the media player **58**. In parallel with this, the data stream is set at step S**8**.**11** into frames of lkB, i.e. there are 1000 bytes between frame start and end points. The media player at step S**8**.**12** calculates the security code (as described in more detail below) and then sends it to the security device **64**, where it is decoded at step S**8**.**13**. On the basis of the decoding, the security device **64** determines at step S**8**.**14** if the security code is valid. If invalid, the security device **64** at step S**8**.**15** resets a timeout counter, thereby preventing a timeout occurring and locking the content. If valid, the memory and interface controller **63** at step S**8**.**16** considers the subsequent data frame as being validated for access. If a valid code is not received before the end of this frame, subsequent frames are filled with random data instead of content data.

[**0419**] The media player **58** recalculates the correct security code once in every frame, but generates 20 security codes for each data frame, 19 of which are incorrect. The media player **58** sends the MMC **56** all the security codes at step S**8**.**17**, in this example resulting in 20 security codes being sent for every frame of data. 19 of these codes are intentionally incorrect, and only one of them is correct. The security device **64** of the MMC **56** compares the results of its calculations with the security code sent by the media player **58**. The security device **64** allows content data to be sent to the player as long as one correct security code is received in every frame. If the security device **64** detects that a valid security code has not been received for a predetermined period of time, using a timer, or if too few codes (either correct or incorrect) are received, then the security device **64** disables access to the data in the non-volatile memory **62**. The security device **64** then needs to be unlocked again by the media player **58** before content playback can be resumed. The security device **64** also locks the MMC **56** if it has not been accessed for a predetermined, configurable period of time.

[**0420**] The security code is calculated based on the following data:

CRC	the last 4 bytes of the decoded validation code (the checksum part)
Bytes	the total number of bytes read from the MMC 56 so far
Random	a number between one half of the number of security updates per frame (in this case, 10 is half of the 20 updates per frame that there are) and 0.

[0421] The media player **58** performs the calculation:

((*CRC*<<Mod 32(Bytes))Xor(Bytes))*Random

[0422] This means that the checksum part (CRC) of the validation code is shifted left by a modulo of 32 of the number of bytes read. The result is Xor-ed with the number of bytes read. The Xor operation consists of applying corresponding bits in the two numbers to respective exclusive-or gates. The result is multiplied by the random number Random.

[0423] The security device **64** in the MMC **56** performs the calculation:

((*CRC*<<Mod 32(Bytes))Xor(Bytes))*Moduloframe size(frame number)

[0424] Moduloframe size(frame number) is frame number modulo 1000 in this instance because the frame size may change, i.e. 1000 e.g. 5032 becomes 0032.

[0425] The result of this is the continual validation of the media player **58** by the security device **64** of the MMC **56**. This prevents it being possible to use a false media player to extract the content data in a useable form; instead the data can only be extracted from the MMC **56** by the correct media player **58**, which renders the content for consumption but does not allow the content data to be used to provide unauthorised copies. The fact that the media player **58** sends many incorrect codes makes it difficult or impossible to determine from examination of the codes sent from the media player **58** to the MMC **56** what calculation is needed to determine the correct codes, thus increasing security since the difficulty of making a false media player which could extract data from the MMC is significantly increased.

[0426] Using these features, the security device **64** is operable to determine whether an external device, comprising the mobile device **40** running the media player **58**, is entitled to access content data from the non-volatile memory **62**, and to allow or disallow access to the content data accordingly.

[0427] An alternative MMC **70** is shown in FIG. **7**. Here, the memory and interface controller **63** is omitted. Instead, a combined memory and interface controller and security device **71** connects the non-volatile memory **62** with the connector pins **61**. This provides the same functionality that the memory and interface controller **63** and the security device **64** do together, but with some additional functionality, as explained below. This embodiment has an advantage in that it could be included within a smaller housing than the FIG. **6** MMC **56**. Since it has less hardware, it may also be less expensive to manufacture. Also, the combined memory and interface controller and security device **71** does not need to support the same type of non-volatile memory as a MMC controller, thereby providing component flexibility.

[0428] The combined memory and interface controller and security device **71** emulates the host interface of a standard MMC controller, so as to allow full connectivity with host devices, such as the mobile device **40**. It also supports additional host interface commands to support security configuration and security validation in some specific hosts. The combined memory and interface controller and security device **71** encrypts all data written to the non-volatile memory **62**, and decrypts all data read from the non-volatile memory **62**. Thus, data accessed by the mobile device **40** is not read from the non-volatile memory **62** directly; instead it is decrypted, processed and buffered in the combined memory and interface controller and security device **71**.

[0429] Some data accessed by the host is a result of processing, for example the security device **64** compiles information for subsequent host requests, or is status information, e.g. security status information, which the media player **58** can use to re-validate security or inform the user of the nature of a problem

[0430] The combined memory and interface controller and security device **71** can be implemented by a microcontroller, an ASIC or an FPGA.

[0431] With the MMCs of both FIGS. **5** and **6**, DRM information is stored in a DRM file within an area of the non-volatile memory **62** which has been defined as a secure area during MMC configuration. The media player **58** can read the DRM file but not influence it, except in the case of a time specific DRM matter. The security device **64** or **71** is arranged to count the number of times that the content is played. If the content is only partially played, this is counted as a play of the content. The number of times that the content has been played is recorded in the DRM file by the security device **64** or **71**. This information can be read by but cannot be not influenced by the media player **58**. The DRM file indicates a maximum number of occasions in which the content data can be played out.

[0432] The DRM data also includes a timeout date or validity date for the content. When the media player **58** is first started, it cooperates with the security device **64** or **71** to write the current time and date of the mobile device **40** from its internal clock (not shown) into the DRM file. If playback of the content is requested and the security device **64** or **71** determines that the latest time and date at which the content could be played has expired (i.e. the current time and date is later than the time/date first recorded plus the validity period), the security validation between the security device **64** or **71** and the media player **58** fails, and an appropriate message is delivered to the user via the display **46**. The same occurs if the limit of the number of occasions on which the content data can be played out is reached. The security device **64** also writes to the DRM file data identifying that the content has expired.

[0433] If after the content has expired once and the time/date of the mobile device **40** is changed to a value that precedes the expiry time/date of the content, the security device **64** or **71** can detect this by detecting data identifying that the content has previously expired in the DRM file. In this case, a predetermined number of further plays of the content, for example 5 plays, are allowed before the content becomes locked requiring a DRM unlock. This is achieved using on-line validation. This feature eases the user impact if the clock in the mobile device was incorrectly configured when the media player **58** was first started.

[0434] The on-line validation process commences with the media player **58** connecting to a DRM server, shown at **80** in FIG. **5**, for example belonging to an entity that is licensed to render content onto MMCs. The DRM server **80** knows the configuration of every MMC **56** that has been released. Connection may be made through WAP or SMS, or in any other suitable way. If the DRM information on the DRM server is valid, the DRM server sends a code through the media player **58** to the security device **64** or **71**, which

causes it to be validated and thus unlocks the content for further playback. This involves updating the DRM file. Locked content can be unlocked again by payment for further content access through a variety of channels (web, wap (e.g. Bango) and SMS).

[0435] If content on an MMC **56**, **71** is locked, the media player **58** will not play the content data back. In this case, the user of the mobile device **40** may arrange for the content to be unlocked for further playback, for example by making an additional payment. This can occur in any suitable way, for example using WAP, a web-based payment service, or by negotiating with an operator by telephone. When payment is made, the DRM server **80** is updated with this information. When the user subsequently starts the media player **58** and attempts to access the locked content, the media player **58** contacts the DRM server **80**, in any suitable way, which sends an unlocking code to the mobile device **40** which the media player **58** passes to the security device **64** or **71**. The security device **64** or **71** then validates the unlocking code, and updates the DRM file to unlock the content.

[0436] Although the mobile device **40** is said to be a mobile (cellular) telephone, it may instead be a personal digital assistant (DA), which may or may not have bidirectional voice communication capabilities. The invention is primarily concerned with providing audio-visual content on a device which is designed primarily for another function. However, the invention is concerned also with dedicated media players.

[0437] Also, although the invention has been described in relation to an MMC **56**, this is not essential. Instead of an MMC, other type of medium including non-volatile memory and an internal memory controller with access to content data stored on the memory being obtained through an interface could be used instead. For example, a memory device with a USB or Bluetooth™ or other interface could be used instead. The housing of the memory device may take any suitable form.

[0438] Where a movie on a DVD is to be provided onto transferable media for use with a general class of target mobile devices, or even where the movie is to be provided for more than a small number of target devices on the same model number, a system such as a system shown in FIG. **9** can be used to advantage. In FIG. **9**, first to third servers **30**, **31**, **32** are shown. The first server **30** is designated as a management node, and includes connections to each of the second and third servers **31**, **32**, which constitute child nodes. Each of the servers **30** to **32** includes at least first and second DVD drives **33**. In this example, DVDs need to be inserted into and extracted from the DVD drives **33** manually, although it is possible to use robots or other automation for this task instead if required.

[0439] Each of the servers **30** to **32** extracts and converts films from DVDs in the DVD drives **33** in parallel. Movies can be extracted from DVDs in a single drive sequentially, i.e. one after the other.

[0440] Assuming sufficient speed for the DVD drive **33** and sufficient processing speed for the servers **30** to **32**, the DVID extraction and conversion process can be completed in respect of one DVD in tens of minutes. Thus, where a serial number of a target device or similar is to be included in the resulting movie to enable the movie to be reproduced only on that target device, the conversion process needs to be effected once for each specific target device. It will be appreciated that the extraction process needs to be performed only once, since the extracted movie is stored in the extracted movie data buffer.

[0441] Where a movie is to be used for a number of target devices of the same class, then the extraction and conversion processes need to be performed only once. Once the movie is stored in mobile format in the mobile format movie data area **20**, it can be copied to an MMC or other removable media device as many times is required. This can be carried out in a suitable manner, for example using internal or external MMC drives.

[0442] The setup for the management system installation specific architecture is in flat files, for example, in a /etc/ subdirectory. The setup for movie production is in database tables using a custom Postgres or Oracle database, although any other suitable database can be used instead, depending on the scale and performance requirements. The management system running on the child node servers **31**, **32** communicate with the management system on the first server **30**. The management node **30** is responsible for task allocation. One instance of the management system is required for each conversion session.

1-52. (canceled)

53. A portable data storage medium comprising:

non-volatile memory having stored therein data comprising computer-readable instructions constituting a media player an interface including terminals for connecting to an external device;

a controller operable to read data out from the non-volatile memory and feed it to the interface; and a security device, in which the media player is operable when running on an external device to interact with the security device in such a way that the security device can determine whether the media player is entitled to access content data from the non-volatile memory, the security device allowing or disallowing access to the content data accordingly.

54. The portable data storage medium of claim 53, wherein a data terminal of the controller is connected to the interface via the security device.

55. The portable data storage medium of claim 53, wherein the security device is integral with the controller.

56. The portable data storage medium of claim 55, wherein the controller and security device is operable to decrypt content data read out from the non-volatile memory.

57. The portable data storage medium of claim 55, wherein the controller is operable also to write data from the interface to the non-volatile memory.

58. The portable data storage medium of claim 53, wherein the media player is operable when running on an external device to detect configuration parameters of the external device, and to control functionality of the media player dependent on the detected configuration parameters.

59. The portable data storage medium of claim 53, wherein when running on an external device, the media player is operable to determine display characteristics of the device, and to select an orientation of video content reproduction dependent on the detected display characteristics.

60. The portable data storage medium of claim 61, wherein the portable data stoppage medium is configured to

control functionality of one or more keys of the external device dependent on the selected orientation.

61. The portable data storage medium of claim 53, wherein the media player is operable to validate the data storage medium by performing a calculation on a validation code received from the data storage medium.

62. The portable data storage medium of claim 53, wherein the security device is operable to determine whether the external device is entitled to access content data by receiving codes and determining that a valid code is received at least once within every predetermined time interval or unit of data.

63. The portable data storage medium of claim 53, wherein the security device is operable to maintain rights management data in the non-volatile memory which is representative of the playback allowability status of the content data.

64. The portable data storage medium of claim 62, wherein the security device is operable to update the rights management data with data identifying a number of occasions on which the content data is played out.

65. The portable data storage medium of claim 62 wherein the security device is operable to store rights management data representative of a time and/or date at which the content data is first played out.

66. The portable data storage medium of claim 53 wherein the security device is responsive to a predetermined data sequence or a predetermined command to enter an active mode.

67. The portable data storage medium of claim 53 wherein the security device is operable to allow or disallow access to content data which is stored in a designated protected area of the non-volatile memory dependent on the determination as to whether the external device is entitled, and to refrain from disallowing access to content data in unprotected areas of the nonvolatile memory.

68. The portable data storage medium of claim 53 wherein the security device is operable to lock the content data on a determination that the validity period of the content data has expired or that a predetermined limit of number of occasions of content data play out is reached, thereby to prevent access to the content data.

69. The portable data storage medium of claim 65 wherein the security device is operable on a determination that the validity period of the content data has expired to update the rights management data with data indicative of content data validity period expiry.

70. The portable data storage medium of claim 68 wherein, in which the security device is responsive to receiving an unlocking code to unlock the content for further playback

71. The portable data storage medium of claim 53 wherein the security device is responsive to receiving the unlocking code to update the rights management data appropriately.

72. The portable data storage medium of claim 53 wherein the security device is responsive to a request for access to content data which is locked to access a remote server through a communication channel of the external device, and to unlock the content data for further playback on receipt of an unlocking code.

73. The portable data storage medium of claim 53 wherein the non-volatile memory has stored therein data comprising computer readable instructions including two or more media players and a loader program, the loader program being operable when running on an external device to determine configuration parameters of the device, to select one of the media players on the basis of the detected configuration parameters, and to control the mobile device to run the selected media player.

* * * * *